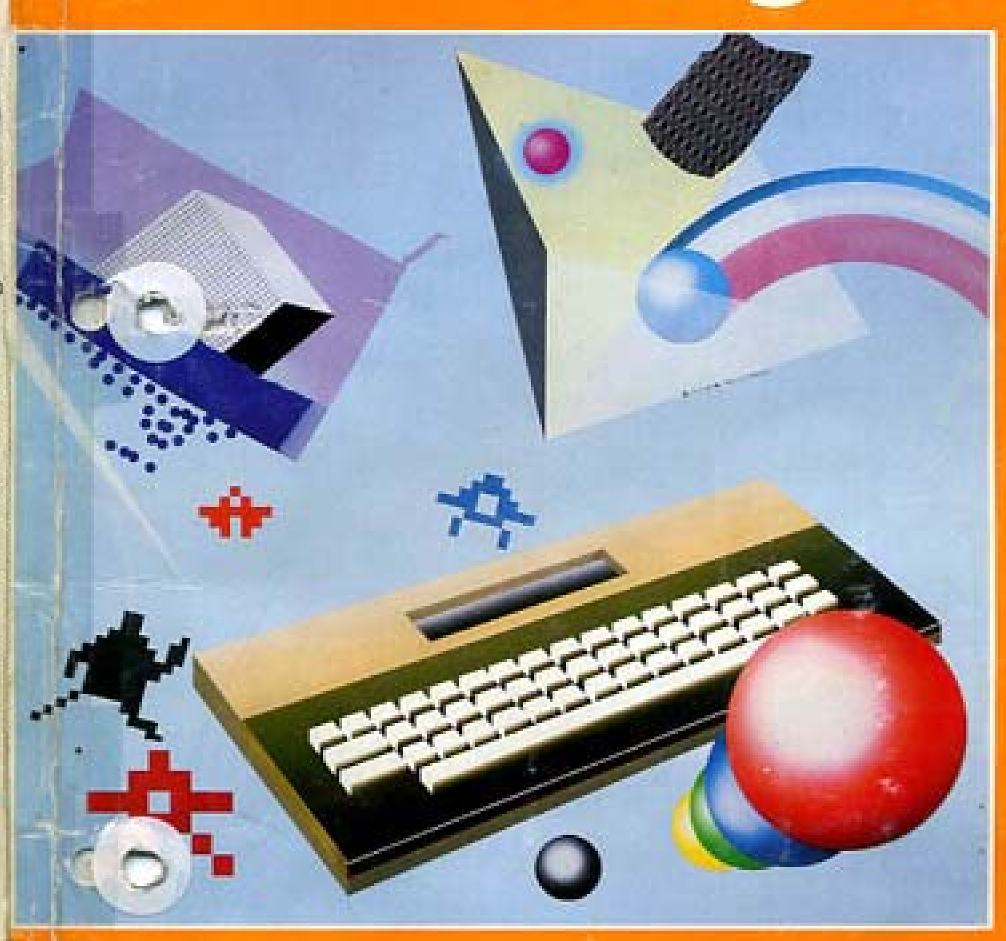
Atari 600 XL/ 800 XL Programme

DM 18,– · 8/80333384

CHIP SPECIAL: Atari 600XL/800XL Programme/Ausgabe 1



Dokumentationen und Listings
für Schule, Beruf und Freizeit
Lernspiele · Wissen · Hobby
Geschicklichkeitsspiele
Mathematik · Grafik

Eine Publikation
aus der Reihe
CHIP-SPECIAL

Mit Programmen „Goldene Diskette '84"

Mathematik	9	Kirschen-Karl
	15	Quadratische Gleichungen
	16	Graph
	19	Nullstelle
Lernen-Wissen-Können	21	Morsen lernen
	23	Sternbilder und Planeten
Abenteuer	30	Princess
Strategiespiel	34	Nuclear
Geschicklichkeit	36	Danger Snake
	78	Verschlungene Pfade
	82	Vier gewinnt
Glücksspiel	37	Jackpot
	41	Kanopus
Kartenspiel	44	Siebzehn und vier
Denkspiel	46	Castle Combat
Merkspiel	48	Töne merken AI/40
	49	Der siebte Sinn für Zahlen AI/20
Mini-Action	50	Saurer Regen
Mini-Strategie	53	X und O
Wirtschaft	55	Finanzmanager
Wissenschaft	57	Sortieren numerischer Felder
Ordnen	61	Programm-Bibliothek
	67	Datenbank
Schreibarbeit	65	Versandaufkleber
Grafik	72	Laterna Magica AI/0
Tips+Tricks	74	Listformat
	76	Fehlermelder
Währung	84	Change
Player-Missile-Grafik	89	Teil 1
		Teil 2
Serviceteil	94	Hilfe zum Erstellen eigener Programme
Die wichtigsten Poke- Befehle	96	Große Peek- und Poke-Liste
Nützliches und Wissenswertes	3	Editorial
	6-8	Programmierhilfe-Karte/Service-Blatt
	103	Buchtips
	104	Alle Atari-User-Clubs in Deutschland
	105	Impressum

Programmierhilfe-Karte

Um Ihnen das Entschlüsseln grafischer Symbole und inverser Zeichen zu erleichtern, sind in den Listings diese Symbole und Zeichen in geschweiften Klammern als Code-Nummer angegeben. Beispielsweise:

{Code 128}

Falls Sie diese ATASCII-Codes und -Zeichen nicht ohnehin schon auswendig können, schauen Sie in nachfolgender Liste nach. So bedeutet zum Beispiel der Code 128:

1. Taste (inverse Video) drücken
2. CONTROL-Taste festhalten
3. gleichzeitig Komma-Taste drücken

Nun erscheint auf dem Bildschirm ein inverses Herz.

Beachte: Zwischenräume, die vor, zwischen oder nach geschweiften Klammern kommen, müssen eingegeben werden. Es können im Listing auch Code-Folgen (z.B. invers geschriebene Wörter) auftauchen.
Beispiel:

Code{200/193/204/204/207}

Vorgehensweis: Wie oben,aber hier sind die Nummern für Folgen der einzelnen Codes lediglich durch Schrägstriche getrennt und werden als Zeichen umgesetzt. Bitte ohne Zwischenraum eingegeben.
Spaces: Wenn {10SPACES} im Listing steht, müssen Sie zehn mal die Leertaste (Spacebar) drücken.
Abkürzungen in der Liste:

CTRL = CONTROL-Taste festhalten
= Inverse Video-Taste drücken (AUF dem Computer = -Taste)
SHIFT = SHIFT-Taste festhalten
(LOWR) = CAPS-Taste drücken bzw. bei alten Geräten LOWR-Taste drücken.

Wenn im Listing folgender Code auftaucht:	Drücken Sie folgende Taste(n):	Auf dem Bildschirm erscheint dann folgendes Zeichen:
0	CTRL-,	
1	CTRL-A	
2	CTRL-B	
3	CTRL-C	
4	CTRL-D	
5	CTRL-E	
6	CTRL-F	
7	CTRL-G	
8	CTRL-H	
9	CTRL-I	
10	CTRL-J	
11	CTRL-K	
12	CTRL-L	
13	CTRL-M	
14	CTRL-N	
15	CTRL-O	
16	CTRL-P	
17	CTRL-Q	
18	CTRL-R	

Wenn im Listing folgender Code auftaucht:	Drücken Sie folgende Taste(n):	Auf dem Bildschirm erscheint dann folgendes Zeichen:
19	CTRL-S	
20	CTRL-T	
21	CTRL-U	
22	CTRL-V	
23	CTRL-W	
24	CTRL-X	
25	CTRL-Y	
26	CTRL-Z	
27	ESC\ESC	
28	ESC\CTRL--	
29	ESC\CTRL-=	
30	ESC\CTRL-+	
31	ESC\CTRL-*	
95	SHIFT-–	
96	CTRL-.	
123	CTRL-;	
124	SHIFT-=	
125	ESC\CTRL-< oder ESC\SHIFT-<	
126	ESC\BACK S	
127	ESC\TAB	
128	(A) CTRL-,	
129	(A) CTRL-A	
130	(A) CTRL-B	
131	(A) CTRL-C	
132	(A) CTRL-D	
133	(A) CTRL-E	
134	(A) CTRL-F	
135	(A) CTRL-G	
136	(A) CTRL-H	
137	(A) CTRL-I	
138	(A) CTRL-J	
139	(A) CTRL-K	
140	(A) CTRL-L	
141	(A) CTRL-M	
142	(A) CTRL-N	
143	(A) CTRL-O	
144	(A) CTRL-P	
145	(A) CTRL-Q	
146	(A) CTRL-R	
147	(A) CTRL-S	
148	(A) CTRL-T	
149	(A) CTRL-U	
150	(A) CTRL-V	
151	(A) CTRL-W	
152	(A) CTRL-X	
153	(A) CTRL-Y	
154	(A) CTRL-Z	
156	ESC\SHIFT-BACK S	
157	ESC\SHIFT->	
158	ESC\CTRL-TAB	
159	ESC\SHIFT-TAB	
160	(A) Leertaste	
161	(A) SHIFT-1	
162	(A) SHIFT-2	
163	(A) SHIFT-3	
164	(A) SHIFT-4	
165	(A) SHIFT-5	
166	(A) SHIFT-6	

Wenn im Listing folgender Code auftaucht:	Drücken Sie folgende Taste(n):	Auf dem Bildschirm erscheint dann folgendes Zeichen:
167	(A) SHIFT-7	'
168	(A) SHIFT-9	(
169	(A) SHIFT-0	)
170	(A) *	*
171	(A) +	+
172	(A) ,	,
173	(A) -	-
174	(A) .	.
175	(A) /	/
176	(A) 0	0
177	(A) 1	1
178	(A) 2	2
179	(A) 3	3
180	(A) 4	4
181	(A) 5	5
182	(A) 6	6
183	(A) 7	7
184	(A) 8	8
185	(A) 9	9
186	(A) SHIFT-;	:
187	(A) ;	;
188	(A) <	<
189	(A) =	=
190	(A) >	>
191		?
192	(A) SHIFT-8	@
193	(A) A	A
194	(A) B	B
195	(A) C	C
196	(A) D	D
197	(A) E	E
198	(A) F	F
199	(A) G	G
200	(A) H	H
201	(A) I	I
202	(A) J	J
203	(A) K	K
204	(A) L	L
205	(A) M	M
206	(A) N	N
207	(A) O	O
208	(A) P	P
209	(A) Q	Q
210	(A) R	R
211	(A) S	S
212	(A) T	T
213	(A) U	U
214	(A) V	V
215	(A) W	W
216	(A) X	X
217	(A) Y	Y
218	(A) Z	Z
219	(A) SHIFT-,	[
220	(A) SHIFT-+	\
221	(A) SHIFT-.	]
		^
	(A) SHIFT--	_

224	(A) CTRL-.
225	(A) (LOWR) A
226	(A) (LOWR) B
227	(A) (LOWR) C
228	(A) (LOWR) D
229	(A) (LOWR) E
230	(A) (LOWR) F
231	(A) (LOWR) G
232	(A) (LOWR) H
233	(A) (LOWR) I
234	(A) (LOWR) J
235	(A) (LOWR) K
236	(A) (LOWR) L
237	(A) (LOWR) M
238	(A) (LOWR) N
239	(A) (LOWR) O
240	(A) (LOWR) P
241	(A) (LOWR) Q
242	(A) (LOWR) R
243	(A) (LOWR) S
244	(A) (LOWR) T
245	(A) (LOWR) U
246	(A) (LOWR) V
247	(A) (LOWR) W
248	(A) (LOWR) X
249	(A) (LOWR) Y
250	(A) (LOWR) Z
251	(A) CTRL-;
252	(A) SHIFT-=
253	ESC\CTRL-2
254	(A) ESC\ CTRL-BACK S
255	(A) ESC\CTRL->

Kirschen-Karl



Beschreibung des Programmes (Atari 800XL, Diskette möglich, 15700 Bytes)

Viel Spaß beim Kopfrechnen vermittelt dieses Mathematik-Lernprogramm für Anfänger, das durch seine hervorragende Grafik auffällt. Nach dem Abtippen des Programmes benötigt der Computer ungefähr eine Minute, bis sämtliche Zeichen umdefiniert und das Maschinenunterprogramm eingeladen ist. Nach der Wartezeit kommt ein kleines Vorspiel, bei dem "Kirschen-Karl" sich die erste Kirsche holt. Nach dem "Karl" oben ist und man auf "Start" gedrückt hat, erscheint die Frage: In welche Klasse gehst du? Man kann die Frage richtig beantworten, der Computer stellt dann den Schwierigkeitsgrad selbst ein. Oder aber man gibt "-1" ein, dann kann man den Schwierigkeitsgrad selber einstellen.

Nun kommt das eigentliche Spiel. Das Programm wurde zum Training von schnellen Kopfrechnern, oder - für Schulanfänger - zum Erlernen der vier Grundrechenarten geschrieben. Gelöst werden sollen die mathematischen Aufgaben möglichst schnell. Es sind, je nach Schwierigkeit, zwei oder drei Zahlen auf dem Bildschirm zu sehen und eine Rechenoperation. Das Ergebnis ist mit Hilfe der Tastatur einzugeben. Nachdem

man auf "Return" gedrückt hat, erscheint die nächste Aufgabe. Hat man alle Aufgaben gelöst, läuft Karl los und holt sich die Kirsche. Ist er zufrieden mit den Ergebnissen, legt er sie bei sich ab. Wenn nicht, wirft er die Kirsche den Drachen zum Fraß vor. Das geht solange, bis "Karl" etwa 20 Kirschen gesammelt hat. Dann ist das Spiel vorbei. Man kann das Spiel auch abbrechen, indem man - während man auf "Return" drückt - bei der Eingabe des Ergebnisses auch auf "Start" drückt. Bei Abbruch durch "Reset" kann es passieren, daß man plötzlich zwei "Karls" hat, von denen einer aber schnell wieder verschwindet. Die Zahlen in den Aufgaben werden vom Zufallsgenerator bestimmt. Somit ist jede Kombination von Aufgaben in den entsprechenden Zahlenbereichen möglich. Die Auswahl der Rechenoperation erfolgt auf die gleiche Weise. Die Division und Subtraktion geht immer auf; es kommen nie negative oder Bruchzahlen als Ergebnis heraus. Die Ansprüche sind hoch gesteckt, damit man nicht auf Anhieb die Forderungen des Computers erfüllt und die Lust verliert. Bringt man das Programm nach Abbruch durch BREAK oder RESET durch RUN wieder zum Laufen, so braucht man die Minute nicht noch einmal abzuwarten; vorausgesetzt, man hat zwischendurch den Computer nicht ausgeschaltet. Viel Spaß!

Jan Spitzkowsky

```
5 REM *KIRSCHEN-KARL*
6 REM *FUER CHIP-SPECIAL*
7 REM *VON JAN SPITZKOWSKY*
10 DIM A(3),A$(42)
15 GRAPHICS 0:? :? :? :? "{4SPACES}Bitte gedulden Sie sich ungefaehr{5SPACES}eine Minute."
20 SP=PEEK(106)-16:POKE 755,3:POKE 756,224
22 IF PEEK(1546)=162 THEN RESTORE 2000:GOTO 50
25 FOR A=0 TO 423:READ B:POKE (SP-2)*256+A,B:NEXT A
30 FOR D=33 TO 58:FOR B=0 TO 7:POKE 256*(SP+8)+D*8+B,PEEK(256*224+D*8+B):NEXT B:NEXT D
32 FOR D=11 TO 14:FOR B=0 TO 7:POKE 256*(SP+8)+D*8+B,PEEK(256*224+D*8+B):NEXT B:NEXT D
35 FOR D=0 TO 59:READ A:FOR B=0 TO 7:READ C:POKE 256*(SP+8)+A*8+B,C:NEXT B:NEXT D
50 DL=PEEK(560)+PEEK(561)*256
55 POKE 54279,SP:PMB=SP*256
60 REM F.A=1023 TO 2048:POKE PMB+A,0:N.A
65 QE=36
70 POKE 82,1
75 POKE 1788,236:POKE 1789,212:POKE 1790,43:POKE 1791,55
80 POKE 559,62:POKE 53277,3
90 POKE 204,SP-2:POKE 203,0:POKE 206,SP+4:POKE 205,0:POKE 209,194:POKE 208,155:POKE 1536,0:POKE 1537,0:POKE 207,20
100 POKE 63,PEEK(88):POKE 64,PEEK(89)
120 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,16,16,16,16,0,24,24,28,0,0
130 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,16,8,4,0,0,20,20,30,0,0
140 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,16,32,64,0,0,36,36,54,0,0
150 DATA 0,0,0,0,0,0,0,0,0,0,1,1,2,16,32,64,128,128,5,167,226,64,0,0
160 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,8,8,8,8,0,24,24,56,0,0
170 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,8,16,32,0,0,40,40,120,0,0
180 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,8,4,2,0,0,36,36,108,0,0
190 DATA 0,0,0,0,0,0,0,0,0,0,128,128,64,8,4,2,1,1,160,234,78,4,0,0
192 DATA 0,0,0,0,0,0,0,0,0,0,0,192,128,131,195,0,0,0,0,36,38,96,0,0
194 DATA 0,0,0,0,0,0,0,0,0,0,0,3,1,193,195,0,0,0,0,36,100,6,0,0
196 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
200 DATA 0,0,0,0,0,60,52,62,52,56,28,24,0,0,0,126,126,0,0,0,0,0,0,0
210 DATA 0,0,0,0,0,44,124,60,12,56,24,24,0,0,0,126,126,0,0,0,0,0,0,0
220 DATA 0,0,0,0,0,60,44,124,44,28,56,24,0,0,0,126,126,0,0,0,0,0,0,0
230 DATA 0,0,0,0,0,52,62,60,48,28,24,24,0,0,0,126,126,0,0,0,0,0,0,0
235 DATA 0,0,0,0,0,60,126,126,126,60,60,24,0,0,0,126,126,0,0,0,0,0,0,0
```

```
240 DATA Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,24,
6Ø,126,Ø,Ø,126,6Ø,Ø,Ø,Ø,Ø,Ø
26Ø DATA Ø,Ø,24,6Ø,255,Ø,Ø,Ø,Ø,Ø,Ø,Ø
,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
28Ø DATA 16,Ø,62,119,119,127,127,127
,62
281 DATA 17,Ø,28,6Ø,28,28,127,127,12
7
282 DATA 18,Ø,62,99,14,56,127,127,12
7
283 DATA 19,Ø,127,14,3Ø,14,111,127,6
2
284 DATA 2Ø,Ø,14,11Ø,11Ø,11Ø,127,3Ø,
3Ø
285 DATA 21,Ø,127,112,126,7,1Ø3,127,
62
286 DATA 22,Ø,63,112,126,119,127,127
,62
287 DATA 23,Ø,127,7,7,14,3Ø,6Ø,6Ø
288 DATA 24,Ø,62,119,62,119,127,127,
62
289 DATA 25,Ø,62,119,63,7,119,127,62
291 DATA 65,8Ø,244,253,253,244,8Ø,Ø,
Ø
292 DATA 66,85,247,247,247,247,85,Ø,
Ø
293 DATA 67,5,31,127,127,31,5,Ø,Ø
294 DATA 68,2Ø,125,125,125,125,125,1
25,125
296 DATA 69,255,255,255,255,255,85,Ø
,Ø
297 DATA 1,Ø,1,4,63,63,63,47,63
298 DATA 3,16,8Ø,32,252,252,188,252,
Ø
3Ø5 DATA Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
3Ø6 DATA 7Ø,128,17Ø,128,128,128,17Ø,
128,128
3Ø7 DATA 71,2,17Ø,2,2,2,17Ø,2,2
3Ø8 DATA 72,255,255,63,63,15,15,3,3
3Ø9 DATA 73,8,8,2ØØ,2ØØ,248,248,252,
252
31Ø DATA 74,128,128,32,32,8,8,1Ø,1Ø
311 DATA 75,Ø,Ø,16Ø,16Ø,8,8,1Ø,1Ø
312 DATA 76,Ø,Ø,Ø,Ø,Ø,48,252,252
313 DATA 77,252,252,48,Ø,Ø,Ø,Ø,Ø
314 DATA 78,128,128,32,4Ø,4Ø,Ø,Ø,Ø
315 DATA 79,255,255,2,24,128,24,2,24
316 DATA 8Ø,128,24,2,24,128,24,2,24
317 DATA 81,255,255,24,152,26,152,26
,24
32Ø DATA 97,Ø,Ø,Ø,Ø,11,3Ø,3Ø,35
321 DATA 98,Ø,Ø,Ø,Ø,Ø,15,255,255
322 DATA 99,Ø,Ø,Ø,8,252,255,255,248
323 DATA 1ØØ,Ø,3,3,3,3,15,15,15
324 DATA 1Ø1,219,219,2Ø7,255,255,255
,252,243
325 DATA 1Ø2,255,253,244,2Ø8,192,Ø,6
4,2Ø8
326 DATA 1Ø3,31,31,31,31,31,3,7,23
327 DATA 1Ø4,243,255,255,255,255,255
,255,255
328 DATA 1Ø5,246,255,255,15,192,192,
Ø,Ø
329 DATA 1Ø6,Ø,8Ø,228,252,24Ø,Ø,Ø,Ø
33Ø DATA 1Ø7,23,31,95,95,95,31,15,15
331 DATA 1Ø8,252,255,255,255,255,255
,255,255
332 DATA 1Ø9,Ø,Ø,64,64,2Ø8,211,211,2
23
333 DATA 11Ø,48,195,2Ø4,2Ø4,24Ø,195,
252,192
334 DATA 111,Ø,Ø,Ø,Ø,1,1,1,Ø
335 DATA 112,31,31,127,127,127,127,1
27,127
336 DATA 113,255,255,252,252,243,243
,243,255
337 DATA 114,63,63,252,252,244,244,1
96,197
338 DATA 115,Ø,Ø,1,5,5,23,23,23
339 DATA 116,255,255,255,255,255,255
,255,255
341 DATA 117,197,53,53,245,245,245,2
53,253
342 DATA 118,Ø,Ø,Ø,Ø,64,64,64,64
343 DATA 119,Ø,Ø,Ø,Ø,Ø,Ø,192,24Ø
344 DATA 12Ø,252,255,255,255,239,62,
255,255
345 DATA 121,Ø,2ØØ,252,191,191,251,2
52,24Ø
346 DATA 122,255,24Ø,Ø,192,192,Ø,Ø,Ø
347 DATA 15,Ø,24,24,Ø,24,24,Ø,Ø
348 DATA 1Ø,Ø,Ø,Ø,24,24,Ø,Ø,Ø
349 DATA 8,129,255,129,129,129,255,1
29,129
35Ø DATA 9,6Ø,66,9Ø,82,82,9Ø,66,6Ø
5ØØ POKE 7Ø4,14:POKE 7Ø5,1Ø:POKE 7Ø6
,88:POKE 7Ø7,54
51Ø C=SP*256:IF PEEK(1546)=162 THEN
53Ø
515 FOR A=1546 TO 1663:READ B:POKE A
,B:NEXT A
52Ø FOR A=Ø TO 661:READ B:POKE C+A,B
:NEXT A
53Ø POKE 12+C,SP
54Ø POKE 2Ø+C,SP
55Ø POKE 28+C,SP
56Ø POKE 36+C,SP+1
565 POKE 59+C,SP+1
568 POKE 74+C,SP+1
57Ø POKE 81+C,SP+2
58Ø POKE 89+C,SP+2
595 GOTO 8ØØ
6ØØ GRAPHICS Ø:? "{CODE29} Bravo. Ve
rsuch doch jetzt mal eine{4SPACES}Kl
```

```
asse hoeher"
6Ø2 POKE 756,224:FOR A=Ø TO 3:POKE 5
3248+A,Ø:NEXT A:? :? :? "Weiter mit
{CODE21Ø/197/212/213/21Ø/2Ø6}"
6Ø3 POKE 2Ø8,Ø:X=USR(C,1Ø8Ø,6):INPUT
 A$:RUN
6Ø4 SETCOLOR 1,Ø,14:TRAP 6Ø4:? "{COD
E125/29}In welche Klasse gehst du";:
INPUT KL:TRAP 4ØØØØ
6Ø5 H=O:GRA=4ØØØ:SC=2Ø:SEK=3.2
6Ø8 IF KL=-1 THEN 76Ø
61Ø IF KL=1 THEN H=5Ø
615 IF KL=2 THEN H=1ØØ
62Ø IF KL=3 THEN H=2ØØ
625 IF KL=4 THEN H=3ØØ
63Ø IF KL=5 THEN H=4ØØ
635 IF KL=6 THEN H=6ØØ
64Ø IF KL=7 THEN H=8ØØ
645 IF KL=8 THEN H=12ØØ
65Ø IF KL>8 THEN H=2ØØØ
66Ø POKE 2Ø7,SC:IF H<3 THEN 6ØØ
68Ø POKE 756,SP+8:POKE 755,1
685 POKE DL+3,68
69Ø FOR D=6 TO 21:POKE DL+D,4:NEXT D
:POKE DL+22,2:POKE DL+23,2:FOR D=24
TO 28:POKE DL+D,6:NEXT D
7ØØ ? "{CODE125/29/29/29/29/3/2/2/2/
2/2/2/2/1}{5SPACES}{CODE3/2/2/2/2/2/
2/2/2/2/2/1/29/29/29/29/3/2/2/2/143/
143/29/3Ø/3Ø/16/16/29/3Ø/3Ø/3Ø/3/2/2
/1/3Ø/28/28/2/2/2/1/3Ø/3Ø/3Ø/28/6/7/
28/3Ø/3Ø/6/7/28/3Ø/3Ø/3Ø/3/2/2/2/1/2
8/28/28/3Ø/3Ø}!#{CODE29/29}"
7Ø2 ? " {CODE6/7/29/3Ø/3Ø/6/7/29/3Ø/
3Ø/6/7/29/3Ø/3Ø/3Ø/3/2/2/2/2/2/2/2/2
/1/29/3Ø/3Ø/3Ø/6/7/29/3Ø/3Ø/6/7/29/3
Ø/3Ø/6/7/29/3Ø/3Ø/6/7/29/3Ø/3Ø/6/7/2
9/3Ø/3Ø/6/7/29/3Ø/3Ø/6/7/29/3Ø/3Ø/3Ø
/3/2/2/2/2/2/2/2/5/28/3Ø/4/29/2/2/2/
2/2/2/2/2/2/2/2/2/2/2/2/2/2/2/2/2/2/
2/1}"
7Ø4 POSITION 1Ø,3
7Ø5 ? "{CODE14Ø/139/29/3Ø/3Ø/136/137
/138/29/3Ø/3Ø/136/137/138/29/3Ø/3Ø/1
36/137/138/29/3Ø/3Ø/136/137/138/29/3
Ø/3Ø/136/137/138/29/3Ø/3Ø/136/137/13
8/29/3Ø/3Ø/136/137/138/29/3Ø/3Ø/136/
137/138/29/3Ø/3Ø/136/137/138/29/3Ø/3
Ø/136/137/142/29/3Ø/3Ø/141}"
72Ø POSITION 1,11
723 ? "{2SPACES}{CODE225/226/227/29/
3Ø/3Ø/3Ø/3Ø/228/229/23Ø/29/3Ø/3Ø/3Ø/
231/232/233/234/29/3Ø/3Ø/3Ø/3Ø/235/2
36/237/238/29/3Ø/3Ø/3Ø/3Ø/3Ø/239/24Ø
/241/242/29/3Ø/3Ø/3Ø/3Ø/243/244/244/
245/246}"
725 SETCOLOR 2,4,4
73Ø SETCOLOR 1,7,1Ø
74Ø POKE 2Ø8,18Ø:POKE 2Ø9,138:POKE 7
56,SP+8
755 GOTO 11ØØ
76Ø ? :? "Wie schnell soll das Maenn
chen sein?{3SPACES}Von 1(schnell) bi
s 255(langsam)":INPUT SC
765 ? :? "Wieviel Sekunden soll man
durch-{7SPACES}schnittlich pro Aufga
be Zeit{11SPACES}haben";:INPUT SEK
768 GRA=SEK*5*5Ø
77Ø ? :? "Wie hoch soll der Zahlenbe
reich{8SPACES}sein, in dem man rechn
et. Von 3 bis{4SPACES}1Ø.ØØØ.ØØØ":IN
PUT H
775 ? :? "Wieviel Strafsekunden bei
einer{8SPACES}falsch beantworteten F
rage";:INPUT SEK
78Ø ? :? "Moechten Sie:":? " - nur z
weistellige Aufgaben .... 3{5SPACES}
- nur dreistellige Aufgaben .... 6"
785 ? " - Strichrechnung dreistellig
 und{9SPACES}Punktrechnung zweistell
ig .... 5":INPUT KL
79Ø GOTO 66Ø
8ØØ ? "{CODE125}":POKE DL+3,68:POKE
756,SP+8
8Ø5 RESTORE 3ØØØ
8Ø6 SETCOLOR 2,4,4:SETCOLOR Ø,1,1Ø
8Ø7 SETCOLOR 1,7,1Ø:SETCOLOR 3,12,4
81Ø FOR A=6 TO 28:POKE DL+A,6:NEXT A
815 FOR A=1Ø TO 28 STEP 5:POKE DL+A,
4:NEXT A:POKE DL+28,4
82Ø POSITION 6,Ø:? "!#":A$="{CODE3/2
/2/2/2/2/2/2/2/2/2/2/2/2/2/2/2/2/2/2
/2/2/2/2/2/2/2/2/2/2/2/1}"
825 FOR A=3 TO 12 STEP 3:POSITION IN
T(RND(Ø)*6)+1,A:? A$:NEXT A:POSITION
 1,14:? A$
83Ø A$="{CODE8/29/3Ø/8}":FOR A=Ø TO
7:READ D,E:POSITION D,E:? A$:NEXT A
84Ø POSITION 5,13:? "{CODE8}":POSITI
ON 25,13:? "{CODE8/28}"
85Ø X=USR(C,579,792,32Ø,8,6)
855 POSITION 1,1Ø:? "VON/ JAN{5SPACE
S}{CODE31}{8SPACES}SPITZKOWSKY":? "{
2SPACES}285Ø BRHV.";
86Ø X=USR(C,264,8Ø8,6ØØ,6)
863 POSITION 26,7:? "UM ZU":POSITION
 6,8:? "BEGINNEN";
865 X=USR(C,8,264,8Ø8,344,6)
867 POSITION 24,4:? "DRUECKEN":? "{C
ODE31/31/31}SIE START,";
87Ø X=USR(C,8Ø8,11,584,8,6)
875 POSITION 5,1:? "WILLKOMMEN{8SPAC
```

```
29/238/173/235/225/242/236}";
880 X=USR(C,260,780,6):POSITION 6,Ø:
? "{2SPACES}":X=USR(C,788,6)
890 FOR A=15 TO -1Ø STEP -Ø.3:D=A:IF
 A<Ø THEN D=Ø
892 SETCOLOR 3,INT(RND(Ø)*3)+6,INT(R
ND(Ø)*6)*2+4:SOUND Ø,1ØØ,6,D
894 IF PEEK(53279)=6 THEN 896
895 NEXT A:GOTO 89Ø
896 SOUND Ø,Ø,Ø,Ø:SETCOLOR 3,12,4:?
"{CODE125}":X=USR(C,364,1144,8,6)
898 SETCOLOR 2,Ø,Ø:FOR A=6 TO 28:POK
E DL+A,2:NEXT A:POKE DL+3,66:POKE 75
6,224:GOTO 6Ø4
11ØØ FOR LAUF=1 TO 5
12ØØ ART=INT(RND(Ø)*4)
121Ø IF (KL=1 AND ART>1) OR (KL=2 AN
D ART=3) THEN 12ØØ
122Ø ON ART+1 GOTO 13ØØ,14ØØ,15ØØ,16
ØØ
13ØØ GOSUB 13Ø5:GOTO 133Ø
13Ø5 FOR A=1 TO 3:A(A)=INT(RND(Ø)*H/
2)
131Ø IF A(A)=Ø THEN A=A-1
132Ø NEXT A:RETURN
133Ø A$="PLUS{4SPACES}+"
135Ø ERG=A(1)+A(2):IF KL>3 THEN ERG=
ERG+A(3)
139Ø GOTO 182Ø
14ØØ GOSUB 13Ø5
141Ø A$="MINUS{3SPACES}-"
142Ø ERG=A(1):A(1)=ERG+A(2):IF KL>3
THEN A(1)=A(1)+A(3)
149Ø GOTO 182Ø
15ØØ GOSUB 151Ø:GOTO 154Ø
151Ø FOR A=1 TO 3:A(A)=INT(RND(Ø)*SQ
R(H))
152Ø IF A(A)=Ø THEN A=A-1
153Ø NEXT A:RETURN
154Ø A$="MAL{5SPACES}*"
155Ø ERG=A(1)*A(2):IF KL>5 THEN ERG=
ERG*A(3)
159Ø GOTO 182Ø
16ØØ GOSUB 151Ø:A$="GETEILT /"
161Ø ERG=A(1):A(1)=ERG*A(2):IF KL>5
THEN A(1)=A(1)*A(3)
169Ø GOTO 182Ø
182Ø IF PEEK(53279)=6 THEN GRAPHICS
Ø:GOTO 6Ø2
183Ø POSITION 18-INT(CLOG(A(1))),19:
? A(1)
184Ø POSITION 38-INT(CLOG(A(2))),19:
? A(2)
185Ø IF (KL>3 AND ART<2) OR KL>5 THE
N POSITION 18-INT(CLOG(A(3))),2Ø:? A
(3)
1855 TRAP 182Ø:POSITION 29-INT(CLOG(
H)),19:IF KL>5 OR (KL>3 AND ART<2) T
HEN ? "{CODE31/31/31/31/31/31/31/31/
31/31/31/31/31/31/31/31/31/31/29}
";
1857 ? A$
186Ø POSITION 17-INT(CLOG(ERG)),21:P
OKE 19,Ø:POKE 2Ø,Ø:INPUT LOE:TRAP 4Ø
ØØØ:Z=PEEK(19)*256+PEEK(2Ø)
1865 IF PEEK(53279)=6 THEN GRAPHICS
Ø:GOTO 6Ø2
187Ø POSITION 1,17:? "{CODE156/156/1
56/156/156/156/156/156}";
1875 T=T+Z:IF LOE<>ERG THEN T=T+SEK*
25Ø:POSITION 1,17:? "LEIDER FALSCH,
VERSUCH ES NOCH EINMAL{CODE253/253/2
53}":GOTO 19ØØ
1877 ? "RICHTIG. ";
1879 IF Z<8Ø THEN ? "BIST DU SICHER,
 DAS DU DIE{4SPACES}RICHTIGE KLASSE
ANGEGEBEN HAST?":GOTO 19ØØ
188Ø IF Z<14Ø THEN ? "DAS HAETTE DEI
N LEHRER AUCH{3SPACES}NICHT SCHNELLE
R GESCHAFFT.":GOTO 19ØØ
1882 IF Z<2ØØ THEN ? "ETWAS LANGSAME
R. ICH KOMM{5SPACES}NICHT MIT.":GOTO
 19ØØ
1883 IF Z<28Ø THEN ? "WENN DAS SO WE
ITERGEHT, HAST{2SPACES}DU DIE KIRSCH
EN BALD VOLL":GOTO 19ØØ
1884 IF Z<36Ø THEN ? "DAS GING JA SC
HON GANZ SCHOEN FIX.":GOTO 19ØØ
1886 IF Z>6ØØ THEN ? "VERSUCH ES DOC
H EINMAL ETWAS{2SPACES}SCHNELLER."
19ØØ NEXT LAUF
191Ø X=USR(C,572,2818,3Ø75,16386,614
7,8193,6151,3Ø1,2674,287,792,262,3Ø8
Ø,528,6)
192Ø X=USR(C,2674,3Ø81,33291,541,11,
132Ø,2722,6)
1925 IF QW=7 THEN QW=Ø:QE=QE-2
193Ø IF T<GRA THEN QW=QW+1:X=USR(C,3
Ø7,8,8,6):POSITION QE,16-QW:? "!#":G
OTO 1985
194Ø X=USR(C,521,2828,2Ø55,316,6):FO
R A=Ø TO 1ØØ:NEXT A
1944 POSITION 12,15
1946 FOR A=Ø TO 2:? "{2SPACES}{CODE2
8/3Ø/3Ø/161}#{CODE3Ø/3Ø}";:GOSUB 198
3:NEXT A
1948 FOR A=Ø TO 3:? "{CODE3Ø/161}# {
CODE3Ø/3Ø/3Ø}";:GOSUB 1983:NEXT A
195Ø ? "{CODE3Ø/161}#{CODE7/3Ø/3Ø/3Ø
}";:GOSUB 1983
1952 ? "{CODE3Ø/161}#{CODE6/3Ø/3Ø/3Ø
}";:GOSUB 1983
1954 ? "{CODE3Ø/161}# {CODE3Ø/3Ø/3Ø}
";:GOSUB 1983
```

```
1970 GOSUB 1984
1972 ? "{CODE226/227/29/30/30/230}!#
{CODE29/30/30/30/233/234}":GOSUB 198
4
1974 ? "{CODE247} {CODE29/30/30/248/
249}#{CODE29/30/30/30/250} ":GOSUB 1
984
1976 ? "{CODE226/227/30/30/29/230}#
{CODE29/30/30/30/233/234}":GOSUB 198
4
1978 ? "{CODE247} {CODE29/30/30/248/
249/29/30/30/250} ":GOSUB 1984
1980 ? "{CODE226/227/29/30/30/230} {
CODE29/30/30/233/234}":GOSUB 1984
1982 GOTO 1985
1983 FOR A1=0 TO PEEK(207)/2:NEXT A1
:RETURN
1984 POSITION 4,11:GOSUB 1983:RETURN
1985 IF T<GRA AND QW=4 AND QE=32 THE
N 600
1988 T=0:POSITION 36,2:? "!#"
1990 GOTO 1100
2000 DATA 162,24,172,3,6,238,3,6,177
,203,172,2,6,238,2,6,145,205,202,208
,237,96
2005 DATA 165,208,160,4,136,153,0,20
8,208,250
2010 DATA 165,209,141,2,6,173,0,6,14
1,3,6,32,10,6
2020 DATA 165,209,141,2,6,173,1,6,14
1,3,6,230,204,230,206,32,10,6
2030 DATA 165,209,141,2,6,230,206,16
9,120,141,3,6,32,10,6
2040 DATA 165,209,141,2,6,230,206,16
9,144,141,3,6,32,10,6
2050 DATA 198,206,198,206,198,206,19
8,204
2060 DATA 169,0,141,3,210
2100 DATA 166,207,160,255,136,208,25
3,202,208,248,96
2300 DATA 104,104,141,246,6,201,1,20
8,4,104,32,100,0,201,2,208,4,104,32,
150,0,201,3,208,4,104,32,200,0
2310 DATA 201,4,208,4,104,32,0,0,201
,6,208,1,96,201,7,240,11,201,11,240,
7,201,8,240,3
2320 DATA 24,144,3,32,50,0
2330 DATA 201,9,240,7,201,10,240,3,2
4,144,4,104,32,202,0
2340 DATA 201,12,208,3,32,0,0,201,5,
208,4,104,32,100,0,169,0,141,1,210,2
4,144,159
2390 DATA 0,0
2400 DATA 141,247,6,230,208,169,0,14
1,1,6,173,0,6,24,105,24,201,24,240,6
,201,48,240,2,169,0,141,0,6
2410 DATA 169,130,141,2,210,169,178,
141,3,210,32,32,6,206,247,6,208,212,
96
2415 DATA 0,0
2420 DATA 141,247,6,198,208,169,48,1
41,1,6,173,0,6,24,105,24,201,120,240
,6,201,144,240,2,169,96,141,0,6
2430 DATA 169,130,141,2,210,169,178,
141,3,210,32,32,6,206,247,6,208,212,
96
2435 DATA 0,0
2440 DATA 141,247,6,198,209,169,96,1
41,1,6,173,0,6,24,105,24
2450 DATA 201,216,240,2,169,192,141,
0,6,32,32,6,206,247,6,208,226,96
2455 DATA 0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0
2460 DATA 141,247,6,230,209,169,96,1
41,1,6,173,0,6,24,105,24
2470 DATA 201,216,240,2,169,192,141,
0,6,32,32,6,206,247,6,208,226,96
2480 DATA 0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0
2500 DATA 173,246,6,201,7,208,11,169
,72,141,0,6,141,1,6,24,144,10
2502 DATA 169,168,141,0,6,169,24,141
,1,6
2504 DATA 165,208,201,174,240,5,169,
50,141,245,6
2506 DATA 169,168,141,1,210,169,4,14
1,247,6
2508 DATA 160,4,185,251,6,153,247,6,
136,208,247
2510 DATA 169,8,141,4,6,174,247,6,12
6,247,6,144,21,173,247,6,56,233,3,14
4,6,198,209,198,209,176,4
2520 DATA 230,209,230,209,24,144,19
2525 DATA 173,246,6,201,7,208,4,230,
208,230,208,201,11,208,4,198,208,198
,208
2527 DATA 32,32,6,238,245,6,238,245,
6,173,245,6,141,0,210
2530 DATA 206,4,6,208,188,206,247,6,
208,178
2540 DATA 169,96,141,0,6,169,48,141,
1,6,32,32,6,96
2700 DATA 141,247,6,169,168,141,1,21
0,165,208,201,129,240,5,169,50,141,2
45,6
2710 DATA 173,246,6,201,9,208,5,198,
209,24,144,2,230,209,32,32,6,238,245
,6,238,245,6,173,245,6,141,0,210
2720 DATA 205,247,6,208,222,96
2800 DATA 165,208,201,174,240,40,201
,190,208,10,160,116,169,0,145,63,200
,145,63,96,165,64,24,105,2
2810 DATA 133,64,160,100,169,1,145,6
3,200,169
```

```
282Ø DATA 3,145,63,165,64,56,233,2,1
33,64,96
283Ø DATA 23Ø,64,16Ø,95,169,Ø,145,63
,2ØØ,145,63,16Ø,135,169,2Ø9,145,63,2
ØØ,145,63
284Ø DATA 169,4,141,245,6,32,32,6,2Ø
6,245,6,2Ø8
285Ø DATA 248,16Ø,95,169,2Ø7,145,63,
2ØØ,145,63
286Ø DATA 16Ø,135,169,2Ø8,145,63,2ØØ
,145,63,198,64,96
29ØØ DATA 141,247,6,169,5Ø,141,245,6
,169,168,141,1,21Ø
29Ø5 DATA 169,72,141,Ø,6,169,72,141,
1,6
291Ø DATA 23Ø,2Ø8,23Ø,2Ø9,23Ø,2Ø9,32
,32,6
292Ø DATA 238,245,6,238,245,6,173,24
5,6,141,Ø,21Ø
295Ø DATA 2Ø6,247,6,2Ø8,23Ø,96
3ØØØ DATA 3,1,23,1,15,4,35,4,4,7,24,
7,14,1Ø,34,1Ø

 READY !
```

Quadratische Gleichungen

Beschreibung des Programmes (Atari 600XL/800XL, 1350 Bytes)

Es gibt in der Praxis eine ganze Reihe von Aufgaben, deren Lösung auf die Lösung einer quadratischen Gleichung hinauslaufen. So zum Beispiel die Beschleunigung eines Autos, der freie Fall eines Fallschirmspringers und Ähnliches. Die allgemeine Form der quadratischen Gleichung lautet:

$Ax^2 + Bx + C = 0$

Dabei sind A, B und C als reelle Zahlen vorausgesetzt. In der Gleichung:

$4x^2 - 7x - 10 = 0$

ist A = 4, B = -7 und C = -10. Man nennt Ax^2 das quadratische, Bx das lineare und C das absolute Glied. Dabei muß $A \neq 0$ sein. Die Lösung der Gleichung erhält man über die Normalform $x^2 + px + q = 0$ mit B/A = p und C/A = q. Die Lösungsformel lautet:

$x_{1/2} = -p/2 \pm \sqrt{(p/2)^2 - q}$

$D = (p/2)^2 - q$ wird die Diskriminante der Normalform genannt. Abhängig von D gibt es drei Lösbarkeitsfälle.

DISKRIMINANTE	WURZEL	BEIDE LÖSUNGEN
D > 0	REELL	REELL, VERSCHIEDEN
D = 0	NULL	REELL, GLEICH
D < 0	IMAGINÄR	KONJUGIERT KOMPLEX

Unser Programm findet alle drei Lösungsformen und gibt sie (auch die konjugiert komplexen) sauber formatiert auf dem Bildschirm aus.

PROGRAMM-AUFBAU

Die Zeilen 100 bis 280 erzeugen das Titelbild. Zeile 290 bis 330: Eingabe der Parameter A, B und C. Zeile 325: FLG ... Flag für die Erkennung einer konjugiert komplexen Lösung. FLG = 0 ... reelle Lösung. FLG = 1 ... komplexe Lösung. Zeile 340 bis 370: Bestimmung von p, q oder D. Zeile 375 bis 430: Ausgabe der Lösungen auf dem Bildschirm. Der imaginäre Lösungsanteil wird mit vorgeschaltetem "j" angegeben, zum Beispiel: x = 3 + j * 2. Zeile 440 bis 480: Programm beenden oder neue Gleichung lösen. Zeile 1000 bis 1100: Dieses Unterprogramm liefert für die Anzeige einer falschen Eingabe einen Mißton.

Mit diesem Programm finden wir für das eingangs angegebene Gleichungsbeispiel $4x^2 - 7x - 10 = 0$ mit A = 4, B = -7 und C = -10 sehr leicht die Lösung:

$x_1 = 2.68210404$, $x_2 = -0.93210404$

J. Peschetz

```
110 DATA ****************************
*
120 DATA **{24SPACES}**
130 DATA ** QUADRATISCHE GLEICHUNG *
*
140 DATA **{24SPACES}**
150 DATA **{4SPACES}FUER CHIP-SPECIA
L{3SPACES}**
160 DATA **{24SPACES}**
170 DATA **{5SPACES}VON J. PESCHETZ{
4SPACES}**
180 DATA **{24SPACES}**
190 DATA ****************************
*
220 DATA ,Die quadratische Gleichung
230 DATA muss folgen Form aufweisen:
240 DATA ,A*X^2 + B*X + C = Ø
250 DATA ,EOF
255 DIM M$(28)
26Ø PRINT CHR$(125):SETCOLOR 2,Ø,Ø
27Ø RESTORE
28Ø READ M$:IF M$<>"EOF" THEN PRINT
M$:GOTO 28Ø
29Ø PRINT :PRINT
3ØØ PRINT "-- A=";:INPUT A:IF A=Ø TH
EN GOSUB 1Ø5Ø:GOTO 3ØØ
31Ø PRINT "-- B=";:INPUT B
32Ø PRINT "-- C=";:INPUT C
33Ø PRINT
335 FLG=Ø
34Ø P=B/A:Q=C/A:P=P/2
35Ø D=P^2 -Q
36Ø IF D<Ø THEN FLG=1
37Ø D=SQR(ABS(D))
375 PRINT "X1=";
38Ø IF FLG=Ø THEN PRINT -P+D:GOTO 4Ø
Ø
39Ø PRINT -P,"+{3SPACES}J*";D
4ØØ PRINT "X2=";
41Ø IF FLG=Ø THEN PRINT -P-D:GOTO 43
Ø
42Ø PRINT -P,"+{3SPACES}J*";-D
43Ø PRINT :PRINT
44Ø PRINT "Neue Gleichung (J/N)"
45Ø A=PEEK(764):IF A=255 THEN 45Ø
46Ø POKE 764,255
47Ø IF A=1 OR A=1Ø6 THEN 26Ø
48Ø END
1Ø5Ø SOUND Ø,12Ø,1Ø,15
1Ø6Ø SOUND 1,125,1Ø,14
1Ø7Ø SOUND 2,255,4,7
1Ø8Ø FOR I=1 TO 5Ø:NEXT I
1Ø9Ø SOUND Ø,Ø,Ø,Ø:SOUND 1,Ø,Ø,Ø:SOU
ND 2,Ø,Ø,Ø
11ØØ RETURN
READY !
```

Graph

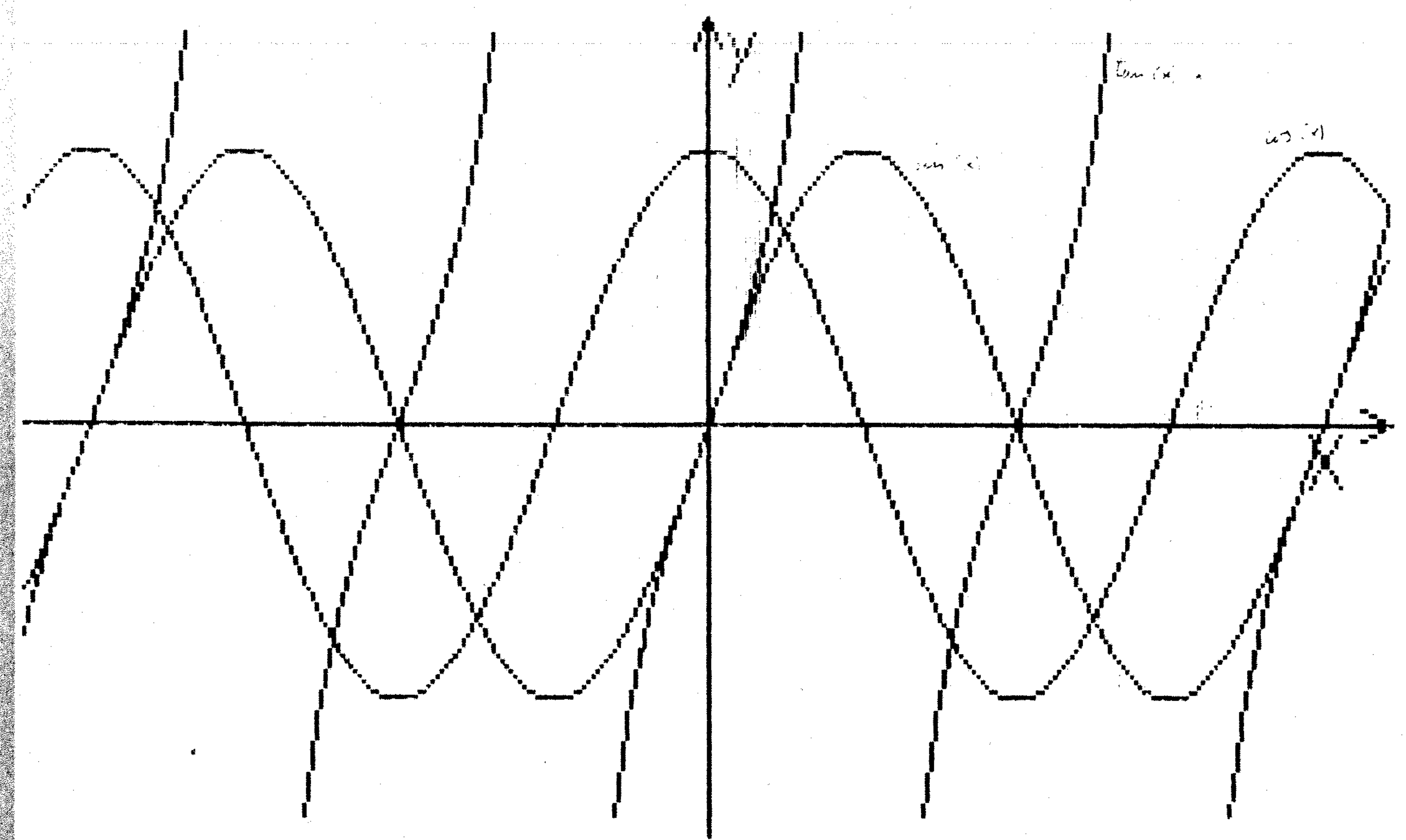
Beschreibung des Programmes (Atari 600XL/800XL/800, 3020 Bytes, Disketten-Einsatz und Druckeranschluß möglich)
Graphen beliebiger Funktionen lassen sich mit diesem Programm zeichnen. Zudem kann man damit komplizierte Nullstellen errechnen. Der Vorteil des Programmes ist, daß es nicht nur rationale Funktionen bis dritten oder vierten Grades behandeln kann, sondern daß es sämtliche Rechenoperationen, die per Computer lösbar sind, beherrscht. Beispiele sind: Sinus, Signum und Logarithmen.
ZUM PROGRAMMABLAUF: Nachdem man das Programm abgetippt und danach RUN eingegeben hat, kann man zum Beispiel mit "1 = " beginnen. Man gibt die Funktion ein und drückt zweimal auf RETURN. Es erscheint die Frage "Von wo bis wo soll die x-Achse gehen?" Das Programm kann diese Entscheidung selber nicht fällen, wo ungefähr der interessante Teil des Graphen liegt. Zuerst wird daher der kleinere Wert eingegeben, und nach jedem Wert wird auf RETURN gedrückt. Danach wird die gleiche Prozedur mit der y-Achse vorgenommen. Bis der Graph gezeichnet ist, dauert es eine Weile. Ist der Graph fertig, gibt es drei Möglichkeiten. Man kann OPTION drücken, und gibt eine zweite Funktion ein, die in denselben Bildausschnitt hineingezeichnet wird, um zum Beispiel eine Sinus- und Cosinus-Funktion zu vergleichen. Man kann SELECT drücken, dann wird das Programm ganz von

vorn gestartet. Dritte Möglichkeit: Der Ausschnitt kann mit START neu gewählt werden, wenn man ihn beim erstenmal nicht getroffen hat. Man kann auch einen Ausschnitt wählen, bei dem eine oder beide Achsen nicht im Bild sind.
Das PROGRAMM-BEISPIEL vergleicht die drei Funktionen: f (x)= sin(x); f (x)= cos(x); f (x)=tan(x)=sin(x) /cos(x).In diesem Programmbeispiel geht die x-Achse von -400 bis 400, die y-Achse von -1,5 bis 1,5. Die Einheiten der Achsen werden durch Hervorheben der Punkte (1;0) und (0;1) im Koordinatensystem kenntlich gemacht. Diese Anwendung ist allerdings nur mit einem im Handel erhältlichen Programm Hardcopy möglich! Als Drucker wurde das Modell Star DP510 verwendet. Da wir in Mathematik dieses Thema gerade in der Schule haben, hat mir dieses Programm schon mehr als einmal geholfen, meine Hausaufgaben schneller und sicherer zu erledigen.

Jan Spitzkowsky

```
1 REM *GRAPH*
2 REM *FUER CHIP-SPECIAL*
3 REM *VON JAN SPITZKOWSKY*
5 GRAPHICS 8:GRAPHICS 40:GRAPHICS 0:
DEG
10 ? :? "Geben Sie mir eine Funktion
 ein{7SPACES}und ich zeichne Ihnen d
en Graphen{5SPACES}dazu."
15 ? "Es gelten fuer die Funktion fo
lgende{2SPACES}Schreibweisen:":? :?
"{2SPACES}'hoch'",,"'^'":? "{2SPACES
}Betrag",,"ABS(...)"
17 ? "{2SPACES}Quadratwurzel","SQR(.
..)"
18 ? "{2SPACES}nat. Log.","LOG(...)"
19 ? "{2SPACES}dek. Log.","CLOG(...)
"
20 ? "{2SPACES}Sinus",,"SIN(...)"
21 ? "{2SPACES}Cosinus","COS(...)"
22 ? :? "Fuer 'x' schreiben Sie ein
grosses X. Nachdem Sie fertig sind,
druecken{5SPACES}Sie so oft auf 'RET
URN',";
23 ? "bis die":? "naechste Frage ers
cheint":?
24 ? "f(X)=";:OPEN #1,4,0,"K:"
25 GET #1,R:IF R=155 THEN CLOSE #1:G
OTO 27
26 ? CHR$(R);:GOTO 25
27 POKE 755,1:POKE 756,PEEK(106)-8:P
OSITION 2,20:? "{CODE255/255}160 YY"
:? "GOTO 30":POKE 764,12:? "{CODE28/
28/28/28/28}":END
30 POKE 755,3:POKE 756,224:IF Q=1 TH
EN 55
31 TRAP 205:? "{CODE125/29/29}Geben
Sie jetzt bitte die Groesse des Grap
hen an. Druecken Sie nach jedem{3SPA
CES}Wert auf 'RETURN'.";
32 ? "Wenn Sie nur":? "auf 'RETURN'
druecken, so erhalten{4SPACES}Sie fu
er die x-Achse die Werte{8SPACES}-20
 und 60";
33 ? " und fuer die":? "y-Achse -10
und 30.":?
39 ? :? "Von wo bis wo soll die x-Ac
hse{8SPACES}gehen";:INPUT XL,XR
40 ? "Von wo bis wo soll die y-Achse
{8SPACES}gehen";:INPUT YU,YO
50 ? "{CODE125}"
55 TRAP 40000:POKE 755,1
60 GRAPHICS 40:COLOR 0:SETCOLOR 2,0,
0
70 Y=160*YO/(YO-YU)-1:X=320*-XL/(XR-
XL)-1
73 ? "{CODE125}"
74 POKE 755,1
75 COLOR 1
80 TRAP 90:PLOT 0,Y:DRAWTO 319,Y:DRA
WTO 312,Y-3:PLOT 319,Y:DRAWTO 312,Y+
3
90 TRAP 100:PLOT X,159:DRAWTO X,0:DR
AWTO X+3,6:PLOT X,0:DRAWTO X-3,6
100 TRAP 110:PLOT 300,Y+2:DRAWTO 307
,Y+12:PLOT 307,Y+2:DRAWTO 300,Y+12
110 TRAP 120:PLOT X+5,3:DRAWTO X+8,8
:PLOT X+11,3:DRAWTO X+5,13
120 X1=320/(XR-XL):Y1=160/(YO-YU)
130 TRAP 135:PLOT X+X1,Y+1:DRAWTO X+
X1,Y-1:PLOT X-1,Y-Y1:DRAWTO X+1,Y-Y1
135 HO=160/ABS(YU-YO)
136 BR=(320/ABS(XR-XL))
138 P=0
140 FOR FX=1 TO 319:TRAP 1000
150 X=FX/BR+XL
160 YY=SQR(4-X*X)
170 FY=-(YY-YO)*HO-1
172 IF FY<0 OR FY>159 THEN P=0:NEXT
FX:GOTO 190
175 IF P=0 THEN P=1:PLOT FX-1,FY
180 DRAWTO FX-1,FY
188 NEXT FX
```

```
190 ? "Der Graph ist fertig.Druecken
 Sie nun:'OPTION'= neue Funktion"
191 ? "'SELECT'= neue Funktion und G
raph{5SPACES}'START' = neuer Graph";
:GOTO 210
205 XL=-20:XR=60:YU=-10:YO=30:GOTO 5
5
210 R=PEEK(53279):IF R=3 THEN GRAPHI
CS 0:Q=1:GOTO 10
220 IF R=5 THEN Q=0:GRAPHICS 8:GRAPH
ICS 40:GRAPHICS 0:GOTO 10
225 IF R=6 THEN GRAPHICS 8:GRAPHICS
40:GRAPHICS 0:GOTO 31
230 GOTO 210
1000 P=0:NEXT FX:GOTO 190
  READY !
```




```
Der Graph ist fertig.Druecken Sie nun:
'OPTION'= neue Funktion
'SELECT'= neue Funktion und Graph
'START' = neuer Graph
```

ATARI PRINTER-INTERFACE READY

Nullstelle einer Funktion

Beschreibung des Programmes (Atari 600XL/800XL, 1400 Bytes)

Nehmen wir an, Ihnen wird folgende Funktion vorgegeben:

$F(x) = 3x^3 - 5x^2 + 7x - 21 = 0$

Wie kann man eine solche Formel lösen? Einen Graph der Funktion zeigt die Abbildung N1. Die Gleichung nennt man kubisch und eine Lösung lautet: x = 2.12181211.

Die Zauberformel für die Lösung heißt: ILLINOIS-ALGORITHMUS Dies ist eine durch M. DOWELL & P. JARRAT ("A modified regula falsi method for computing the root of an equation", BIT 11/1971, S. 168-174) gefundene Verbesserung der als - REGULA FALSI bekannten Iteration, um die Wurzel einer Gleichung aufzufinden. Hierbei ersetzt man nach Abbildung N2 die Kurve im Intervall x ,x durch die Sekante (im Sinn einer linearen Interpolation).

Der neue Schnittpunkt x dieser Sekante mit der X-Achse ist der Ausgangspunkt für die nächste Iteration. Die neuen Parameter für die nächste Iteration werden immer so gewählt, daß die Funktionswerte umgekehrte Vorzeichen aufweisen. Ist der Funktionswert kleiner als ein vorgegebener Grenzwert, wird die Iteration abgebrochen und die Nullstelle x ist gefunden.

ITERATIONSVORSCHRIFT:

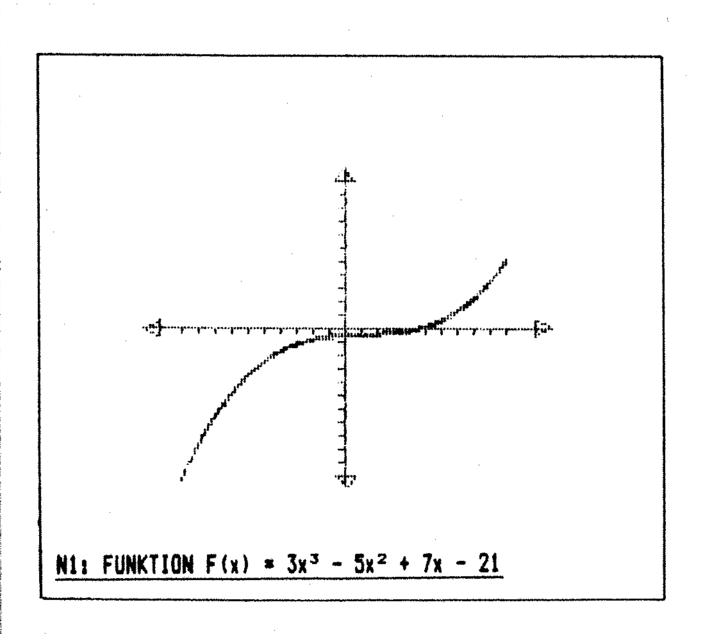
x = x -(x -x)/(F(x)-F(x))*F(x)

Diese Formel berechnet den Schnittpunkt der Geraden durch die Punkte (x ,F(x)),(x ,F(x)) mit der X-Achse (Abbildung N2)

PROGRAMM-AUFBAU

In der Zeile 250 muß die Funktion, deren Nullstelle bestimmt werden soll, definiert werden. Zeile 290 bis 430: Eingabe des Intervalles X1, X2, in dem die Nullstelle vermutet wird. Die Startwerte müssen so gewählt werden, daß sich unterschiedliche Funktionswerte ergeben: Zeile 450 bis 580: Dieser Programmteil findet die Nullstelle. Zeile 600 bis 630: Gibt die Nullstelle aus und berechnet zur Kontrolle den Funktionswert FX3 mit X3 als Nullstelle. Zeile 640 bis 680: Programmabbruch oder Suchen einer weiteren Nullstelle!

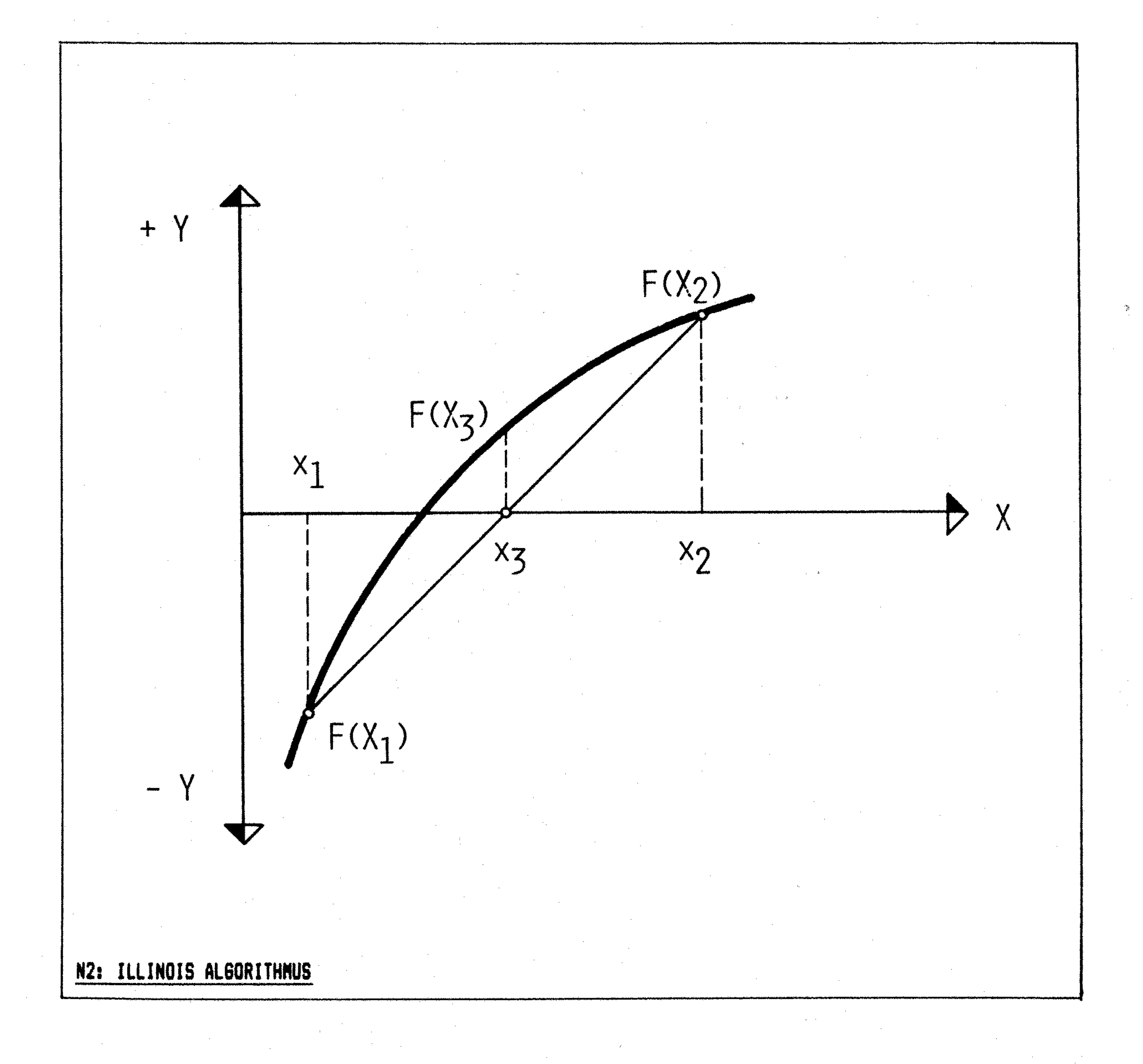
J. Peschetz



N1: FUNKTION $F(x) = 3x^3 - 5x^2 + 7x - 21$

```
100 REM *NULLSTELLE EINER FUNKTION*
110 REM *FUER CHIP-SPECIAL*
120 REM *VON J.PESCHETZ*
245 DIM NAME$(40):GOTO 290
250 FX=3*X^3-5*X^2+7*X-21:RETURN
290 PRINT CHR$(125):SETCOLOR 2,10,0
300 PRINT "NULLSTELLENBESTIMMUNG DUR
CH":PRINT "REGULA FALSI (ILLINOIS AL
GORITHMUS)":PRINT :PRINT
310 PRINT "Ist die Funktion F(X) unt
er":PRINT "Zeile Nr. 250 definiert (
J/N?)":PRINT :PRINT
320 A=PEEK(764):IF A=255 THEN 320
330 POKE 764,255
340 IF A=1 OR A=106 THEN 360
350 STOP
360 PRINT CHR$(255)
370 PRINT "FUNKTIONSNAME ":INPUT NAM
E$
380 PRINT :PRINT "Die Startwerte X1
und X2 sind so zu "
385 PRINT "waehlen, dass die Funktio
```

```
nswerte ent-"
39Ø PRINT "gegengesetzte Vorzeichen
aufweisen!":PRINT :PRINT
4ØØ PRINT "STARTWERTE X1,X2 ":INPUT
X1,X2
41Ø X=X1:GOSUB 25Ø:FX1=FX
42Ø X=X2:GOSUB 25Ø:FX2=FX
43Ø IF FX1*FX2>Ø THEN 38Ø
47Ø X3=X2-(X2-X1)/(FX2-FX1)*FX2
48Ø X=X3:GOSUB 25Ø:FX3=FX
49Ø IF FX3=Ø THEN 59Ø
5ØØ IF ABS(FX3)<1.ØE-Ø5 THEN 62Ø
51Ø IF FX3*FX2>Ø THEN 55Ø
52Ø X1=X2
53Ø FX1=FX2
54Ø GOTO 56Ø
55Ø FX1=FX1/2
56Ø X2=X3
57Ø FX2=FX3
58Ø GOTO 47Ø
62Ø PRINT :PRINT "Die Nullstelle von
":PRINT NAME$:PRINT "lautet:":PRINT
:PRINT "X=";X3
625 X=X3:GOSUB 25Ø
63Ø PRINT :PRINT "Kontrolle:":PRINT
:PRINT "F(";STR$(X3);")=";FX
64Ø PRINT :PRINT "Neue Werte (J/N)?"
65Ø A=PEEK(764):IF A=255 THEN 65Ø
66Ø POKE 764,255
67Ø IF A=1 OR A=1Ø6 THEN 38Ø
68Ø END
  READY !
```



N2: ILLINOIS ALGORITHMUS

Morsen lernen

Beschreibung des Programmes (Atari 600XL/800XL/400/800,5250 Bytes)
Dieses Programm gibt Ihnen die Möglichkeit, das Morsealphabet zu lernen und ganze Sätze vom Computer morsen zu lassen. Nach dem Eintippen haben Sie auf der Titelseite einen Überblick über das Morsealphabet; Sie können einen Text eingeben. Danach haben Sie die Wahl zum gemorsten Text die entsprechenden Buchstaben oder nur die Morsezeichen zu sehen. Außerdem können Sie wählen mit welcher Geschwindigkeit Ihr Text gemorst wird. Und zwar zwischen drei und 20 Zeichen in der Minute. Am Ende gibt Ihnen das Programm die Wahl, einen neuen Text einzugeben oder die Morserei zu beenden.

Bernd Moser

```
10 REM *MORSEN LERNEN*
11 REM *FUER CHIP-SPECIAL*
12 REM *VON BERND MOSER*
19 GOTO 5000
20 FOR PAUSE=1 TO 500:NEXT PAUSE:RET
URN
50 ? ".";:SOUND 0,40,10,12:FOR J=1 T
O X:NEXT J
51 SOUND 0,0,0,0:FOR J=1 TO X:NEXT J
60 RETURN
70 ? "-";:SOUND 0,40,10,12:FOR J=1 T
O T:NEXT J
71 SOUND 0,0,0,0:FOR J=1 TO X:NEXT J
80 RETURN
90 L=LEN(A$):FOR I=1 TO L:R$=A$(I,I)
:IF R$="K" THEN GOSUB 50
100 IF R$="L" THEN GOSUB 70
110 NEXT I:RETURN
120 ? :? :? :GOSUB 20
210 FOR G=1 TO 0
220 ZZ=LEN(TEXT$):FOR QQ=1 TO ZZ:JN$
=TEXT$(QQ,QQ)
230 BM=ASC(JN$)
240 IF BM>57 AND BM<63 THEN NEXT QQ
245 IF BM=32 THEN ? CHR$(BM):FOR I=1
 TO CH:NEXT I:NEXT QQ
250 IF BM=64 THEN NEXT QQ
260 IF BM=45 THEN NEXT QQ
265 IF BM<44 THEN NEXT QQ
266 IF BM>99 THEN NEXT QQ
270 GOSUB BM*10
280 IF HI=1 THEN ? CHR$(BM);"{2SPACE
S}";
290 GOSUB 90
293 ?
300 FOR I=1 TO CH:NEXT I
310 NEXT QQ
330 ? :FOR D=1 TO HC:NEXT D
340 NEXT G
350 POSITION 0,22:? "{4SPACES}Wollen
 Sie aufhoeren? (J/N)"
351 POSITION 16,23:POKE 752,0:INPUT
JN$
360 IF JN$="J" THEN POKE 82,2:END
370 POKE 752,1:? CHR$(125):POSITION
2,4
381 POKE 752,1:POKE 82,0:GOTO 5040
390 GOTO 6000
440 A$="LLKKLL":RETURN :REM *,*
460 A$="KLKLKL":RETURN :REM *.*
470 A$="LKKLK":RETURN :REM */*
480 A$="LLLLL":RETURN :REM *0*
490 A$="KLLLL":RETURN :REM *1*
500 A$="KKLLL":RETURN :REM *2*
510 A$="KKKLL":RETURN :REM *3*
520 A$="KKKKL":RETURN :REM *4*
530 A$="KKKKK":RETURN :REM *5*
540 A$="LKKKK":RETURN :REM *6*
550 A$="LLKKK":RETURN :REM *7*
560 A$="LLLKK":RETURN :REM *8*
570 A$="LLLLK":RETURN :REM *9*
630 A$="KKLLKK":RETURN :REM *?*
650 A$="KL":RETURN :REM *A*
660 A$="LKKK":RETURN :REM *B*
670 A$="LKLK":RETURN :REM *C*
680 A$="LKK":RETURN :REM *D*
690 A$="K":RETURN :REM *E*
700 A$="KKLK":RETURN :REM *F*
710 A$="LLK":RETURN :REM *G*
720 A$="KKKK":RETURN :REM *H*
730 A$="KK":RETURN :REM *I*
740 A$="KLLL":RETURN :REM *J*
750 A$="LKL":RETURN :REM *K*
760 A$="KLKK":RETURN :REM *L*
770 A$="LL":RETURN :REM *M*
780 A$="LK":RETURN :REM *N*
790 A$="LLL":RETURN :REM *O*
800 A$="KLLK":RETURN :REM *P*
810 A$="LLKL":RETURN :REM *Q*
820 A$="KLK":RETURN :REM *R*
830 A$="KKK":RETURN :REM *S*
840 A$="L":RETURN :REM *T*
```

```
850 A$="KKL":RETURN :REM *U*
860 A$="KKKL":RETURN :REM *V*
870 A$="KLL":RETURN :REM *W*
880 A$="LKKL":RETURN :REM *X*
890 A$="LKLL":RETURN :REM *Y*
900 A$="LLKK":RETURN :REM *Z*
5000 GRAPHICS 0:POKE 710,0:POKE 709,
14
5001 CLR :DIM ALPH$(15),A$(6),JN$(1)
,R$(1),TEXT$(100)
5002 POKE 82,0
5003 POSITION 12,0:? "MORSE ALPHABET
":POSITION 12,1
5004 FOR I=1 TO 14:? CHR$(13);:NEXT
I:?
5005 ? " Dieses Programm morst einen
 von Ihnen"
5006 ? "{2SPACES}eingegebenen Text i
n verschiedenen"
5007 ? "{3SPACES}Geschwindigkeiten."
:?
5008 ? CHR$(17);:FOR OFT=1 TO 38:? C
HR$(18);:NEXT OFT:? CHR$(5)
5009 FOR I=7 TO 16:POSITION 0,I:? CH
R$(124)
5010 POSITION 39,I:? CHR$(124):NEXT
I
5011 ? CHR$(28);CHR$(26);:FOR OFT=1
TO 38:? CHR$(18);:NEXT OFT:? CHR$(3)
5023 POSITION 1,7:? "0=----- *A=.-{4
SPACES}*K=-.-{3SPACES}*U=..-{2SPACES
}"
5024 POSITION 1,8:? "1=.---- *B=-...
{2SPACES}*L=.-..{2SPACES}*V=...- "
5025 POSITION 1,9:? "2=..--- *C=-.-.
{2SPACES}*M=--{4SPACES}*W=.--{2SPACE
S}"
5026 POSITION 1,10:? "3=...-- *D=-..
{3SPACES}*N=-.{4SPACES}*X=-..- "
5027 POSITION 1,11:? "4=....- *E=.{5
SPACES}*O=---{3SPACES}*Y=-.-- "
5028 POSITION 1,12:? "5=..... *F=..-
.{2SPACES}*P=.--.{2SPACES}*Z=--.. "
5029 POSITION 1,13:? "6=-.... *G=--.
{3SPACES}*Q=--.-{2SPACES}*"
5030 POSITION 1,14:? "7=--... *H=...
.{2SPACES}*R=.-.{3SPACES}*,=--..--"
5031 POSITION 1,15:? "8=---.. *I=..{
4SPACES}*S=...{3SPACES}*.=.-.-.-"
5032 POSITION 1,16:? "9=----. *J=.--
-{2SPACES}*T=-{5SPACES}*?=..--.."
5040 POSITION 4,19:? "Geben Sie den
Text ein:{4SPACES}":POSITION 0,20:PO
KE 752,0:INPUT TEXT$
5050 POKE 752,1:? CHR$(125):POSITION
 4,4
5051 ? "Wollen Sie den Text sehen? (
J/N)"
5052 POSITION 16,5:POKE 752,0:INPUT
JN$:IF JN$(1,1)<>"J" THEN HI=0:GOTO
5060
5053 HI=1
5060 POKE 752,1:? CHR$(125):POSITION
 2,4
5061 ? "Welche Geschwindigkeit wolle
n Sie?"
5062 ? "{2SPACES}(3-20 Worte pro Min
ute)?"
5063 ? "{10SPACES}";:INPUT TEMPO
5080 IF TEMPO<3 OR TEMPO>20 THEN 506
0
5085 IF HI=0 THEN POKE 752,1
5090 ? CHR$(125):X=34:T=X*3:CH=0:HC=
0:IF TEMPO=3 THEN CH=170:HC=1020:GOT
O 6000
5100 IF TEMPO=4 THEN CH=130:HC=750:G
OTO 6000
5110 IF TEMPO=5 THEN CH=102:HC=612:G
OTO 6000
5120 IF TEMPO=6 THEN CH=85:HC=510:GO
TO 6000
5130 IF TEMPO=7 THEN CH=73:HC=435:GO
TO 6000
5140 IF TEMPO=8 THEN CH=64:HC=382:GO
TO 6000
5150 IF TEMPO=9 THEN CH=57:HC=340:GO
TO 6000
5160 IF TEMPO=10 THEN CH=51:HC=306:G
OTO 6000
5170 IF TEMPO=11 THEN CH=46:HC=275:G
OTO 6000
5180 IF TEMPO=12 THEN CH=42:HC=255:G
OTO 6000
5190 IF TEMPO=13 THEN CH=39:HC=235:G
OTO 6000
5200 IF TEMPO=14 THEN CH=36:HC=218:G
OTO 6000
5210 IF TEMPO=15 THEN CH=X:HC=T*2:GO
TO 6000
5220 X=30.5:T=X*3
5230 IF TEMPO=16 THEN CH=34:HC=205:G
OTO 6000
5240 IF TEMPO=17 THEN CH=32:HC=193:G
OTO 6000
5250 IF TEMPO=18 THEN CH=X:HC=T*2:GO
TO 6000
5260 X=27:T=X*3
5270 IF TEMPO=19 THEN CH=29:HC=170:G
OTO 6000
5280 IF TEMPO=20 THEN CH=X:HC=T*2:GO
TO 6000
6000 POKE 82,10:? "TEXT WIRD GEMORST
:":? :GOTO 120
  READY !
```

Sternbilder und Planeten

Beschreibung des Programmes (Atari 600XL/800XL, 21500 Bytes)
Kennen Sie sich mit den einzelnen Sternbildern und Planeten aus? Dieses Programm verschafft Ihnen einen Überblick. Sobald Sie das Listing eingegeben haben, erscheint nach dem Titel das erste Menü. Sie können zwischen Sternbildern oder Planeten wählen.
Menü Sternbilder: Hier haben Sie die Wahl zwischen 16 verschiedenen Sternbildern. Sobald Sie die entsprechende Nummer eingeben erscheint das Sternbild. Wenn Sie auf "J" tippen werden Ihnen weitere Informationen zu dem gewählten Sternbild vermittelt. Mit der Nummer 17 kommen Sie zum ...
Menü Planeten:
Auch hier geben Sie lediglich die Nummer des gewünschten Planeten ein. Auf dem Bildschirm erscheint dann ein grafischer Größenvergleich des gewählten Planeten mit der Erde. Auch hier können Sie weitere Informationen mit "J" abrufen. Die Nummer elf führt Sie ins Sternbildermenü zurück.

Bernd Moser

```
100 REM * STERNE *
101 REM * FUER CHIP-SPECIAL*
102 REM * VON BERND MOSER *
170 GRAPHICS 1:SETCOLOR 2,Ø,Ø:SETCOL
OR Ø,Ø,15
180 DIM STERN$(35),PLANET$(35),PLANE
T2$(35)
190 FOR OFT=1 TO 5Ø
200 SOUND Ø,INT(RND(Ø)*75),1Ø,8
210 S1=INT(2Ø*RND(Ø))
220 S2=INT(2Ø*RND(Ø))
230 POSITION S1,S2:? #6;CHR$(46)
240 FOR WA=1 TO 25:NEXT WA
250 NEXT OFT
260 SOUND Ø,Ø,Ø,Ø
270 FOR OFT=Ø TO 15
280 SOUND 2,OFT,8,5:FOR WA=1 TO 3Ø:N
EXT WA
290 SETCOLOR Ø,Ø,OFT
300 POSITION 3,5:? #6;"PLANETEN UND"
:POSITION 3,7:? #6;"STERNBILDER"
310 POSITION 3,12:? #6;"(C) 1983 BY"
:POSITION 3,14:? #6;"BERND MOSER":PO
SITION 3,16:? #6;"WEST GERMANY"
320 FOR WA=1 TO 2ØØ:NEXT WA
330 NEXT OFT
340 FOR OFT=15 TO Ø STEP -1
350 SOUND 2,OFT,8,5:FOR WA=1 TO 1ØØ:
NEXT WA
360 NEXT OFT:SOUND 2,Ø,Ø,Ø
410 GRAPHICS Ø:SETCOLOR 2,Ø,Ø:POKE 7
52,1
420 FOR X=1 TO 2Ø:POSITION X,1:? "*"
:POSITION X,2Ø:? "*":FOR WAIT=1 TO 1
5:NEXT WAIT:NEXT X
430 FOR Y=1 TO 2Ø:POSITION 1,Y:? "*"
:POSITION 2Ø,Y:? "*":FOR WAIT=1 TO 1
5:NEXT WAIT:NEXT Y
440 POSITION 3,3:FOR I=1 TO 3:? CHR$
(16Ø);:NEXT I:POSITION 3,4:? CHR$(16
Ø)
450 POSITION 3,5:FOR I=1 TO 3:? CHR$
(16Ø);:NEXT I:POSITION 5,6:? CHR$(16
Ø)
460 POSITION 3,7:FOR I=1 TO 3:? CHR$
(16Ø);:NEXT I
470 POSITION 3,1Ø:FOR I=1 TO 3:? CHR
$(16Ø);:NEXT I
480 POSITION 3,11:? CHR$(16Ø):POSITI
ON 5,11:? CHR$(16Ø)
490 POSITION 3,12:FOR I=1 TO 3:? CHR
$(16Ø);:NEXT I:POSITION 3,13:? CHR$(
16Ø):POSITION 3,14:? CHR$(16Ø)
500 POSITION 6,7:? "TERNBILDER":POSI
TION 4,14:? "LANETEN"
510 POSITION 3,17:? "DRUECKEN SIE 'S
'":POSITION 3,18:? "ODER 'P'....."
520 CLOSE #1:OPEN #1,4,Ø,"K"
530 GET #1,A
540 IF A=ASC("S") THEN 6ØØ
550 IF A=ASC("P") THEN 31ØØ
560 GOTO 41Ø
600 GRAPHICS Ø:SETCOLOR 2,Ø,Ø:POKE 7
52,1
610 ? "{7SPACES}S T E R N B I L D E
R":? :? "{1ØSPACES}Waehlen Sie aus:"
620 ? :? " 1. Grosser Wagen{4SPACES}
2. Jungfrau":?
630 ? " 3. Bootes{11SPACES}4. Loewe"
:?
640 ? " 5. Krebs{12SPACES}6. Stier":
?
```

```
65Ø ? " 7. Fuhrmann{9SPACES}8. Perse
us":?
66Ø ? " 9. Schwan{1ØSPACES}1Ø. Adler
":?
67Ø ? "11. Andromeda{7SPACES}12. Cas
siopeia":?
68Ø ? "13. Pegasus{9SPACES}14. Orion
":?
69Ø ? "15. Wassermann{6SPACES}16. He
rkules":?
7ØØ ? "{9SPACES}17. Planeten"
71Ø ? :INPUT STERN:IF STERN<Ø OR STE
RN>17 THEN 6ØØ
72Ø STERN=INT(STERN)
73Ø IF STERN=Ø THEN GOSUB 3Ø4Ø:GOTO
6ØØ
74Ø IF STERN=17 THEN GOTO 31ØØ
79Ø GRAPHICS 8:SETCOLOR 2,Ø,Ø:SETCOL
OR Ø,Ø,15:COLOR 1
8ØØ POKE 752,1
8Ø1 IF STERN=1 THEN RESTORE 113Ø
8Ø2 IF STERN=2 THEN RESTORE 117Ø
8Ø3 IF STERN=3 THEN RESTORE 121Ø
8Ø4 IF STERN=4 THEN RESTORE 129Ø
8Ø5 IF STERN=5 THEN RESTORE 133Ø
8Ø6 IF STERN=6 THEN RESTORE 137Ø
8Ø7 IF STERN=7 THEN RESTORE 141Ø
8Ø8 IF STERN=8 THEN RESTORE 145Ø
8Ø9 IF STERN=9 THEN RESTORE 151Ø
81Ø IF STERN=1Ø THEN RESTORE 155Ø
811 IF STERN=11 THEN RESTORE 159Ø
812 IF STERN=12 THEN RESTORE 163Ø
813 IF STERN=13 THEN RESTORE 166Ø
814 IF STERN=14 THEN RESTORE 17ØØ
815 IF STERN=15 THEN RESTORE 175Ø
816 IF STERN=16 THEN RESTORE 179Ø
82Ø READ STERN$
83Ø READ X,Y:IF X=-1 THEN 96Ø
84Ø PLOT X,Y:DRAWTO X+1Ø,Y
85Ø PLOT X,Y+1:DRAWTO X+1Ø,Y+1
86Ø PLOT X,Y+2:DRAWTO X+1Ø,Y+2
87Ø PLOT X,Y+3:DRAWTO X+1Ø,Y+3
88Ø PLOT X,Y+4:DRAWTO X+1Ø,Y+4
89Ø PLOT X,Y+5:DRAWTO X+1Ø,Y+5
9ØØ PLOT X,Y+6:DRAWTO X+1Ø,Y+6
91Ø PLOT X,Y+7:DRAWTO X+1Ø,Y+7
92Ø PLOT X,Y+8:DRAWTO X+1Ø,Y+8
93Ø PLOT X,Y+9:DRAWTO X+1Ø,Y+9
94Ø PLOT X,Y+1Ø:DRAWTO X+1Ø,Y+1Ø
95Ø GOTO 83Ø
96Ø READ X,Y:IF X=-1 THEN 1Ø1Ø
97Ø PLOT X,Y:DRAWTO X+3,Y
98Ø PLOT X,Y+1:DRAWTO X+3,Y+1
99Ø PLOT X,Y+2:DRAWTO X+3,Y+2
1ØØØ PLOT X,Y+3:DRAWTO X+3,Y+3:GOTO
96Ø
1Ø1Ø READ X,Y,X1,Y1:IF X=-1 THEN 1Ø4
Ø
1Ø2Ø PLOT X,Y:DRAWTO X1,Y1
1Ø3Ø GOTO 1Ø1Ø
1Ø4Ø ? STERN$:? :? "WEITERE INFORMAT
IONEN (J/N)"
1Ø5Ø CLOSE #2:OPEN #2,4,Ø,"K"
1Ø6Ø GET #2,A
1Ø7Ø IF A=ASC("J") THEN 186Ø
1Ø8Ø GOTO 6ØØ
113Ø DATA GROSSER WAGEN / GROSSER BA
ER,256,48,244,1Ø4,156,1ØØ,152,52,1Ø4
,4Ø,64,32,24,44,-1,-1
114Ø DATA -1,-1
115Ø DATA 264,56,252,112,252,112,164
,1Ø8,164,1Ø8,16Ø,6Ø,16Ø,6Ø,264,56,16
Ø,6Ø,112,48,112,48,72,4Ø,72,4Ø,32,52
116Ø DATA -1,-1,-1,-1
117Ø DATA JUNGFRAU,1ØØ,44,2ØØ,44,2Ø4
,84,244,4,3ØØ,124,64,144,-1,-1
118Ø DATA 132,92,64,4,244,1Ø4,3ØØ,1Ø
Ø,3ØØ,1Ø5,3Ø8,1Ø8,-1,-1
119Ø DATA 1Ø8,52,72,12,1Ø8,52,2Ø8,52
,2Ø8,52,252,12,2Ø8,52,212,92,212,92,
3Ø8,132,212,92,132,92,132,92,72,152
12ØØ DATA 132,92,1Ø8,52,-1,-1,-1,-1,
121Ø DATA BOOTES,228,24,224,84,144,1
Ø4,84,124,24,84,76,44,-1,-1
122Ø DATA 24Ø,12,25Ø,12
123Ø DATA 236,124,244,13Ø
124Ø DATA 224,14Ø,2Ø4,148
125Ø DATA 144,76,128,68
126Ø DATA 36,132,-1,-1
127Ø DATA 232,92,236,32,232,92,232,1
24,232,92,152,112,152,112,92,132,92,
132,32,96,32,96,36,132,32,96,84,52,8
4,52
128Ø DATA 232,92,-1,-1,-1,-1
129Ø DATA LOEWE,184,64,216,96,224,12
4,4,144,84,64,84,124,184,24,244,2Ø,-
1,-1
13ØØ DATA 276,14Ø,124,68,228,12,-1,-
1
131Ø DATA 184,72,224,1Ø4,224,1Ø4,232
,132,232,132,276,14Ø,232,132,92,132,
92,132,12,152,12,152,92,72,92,72,184
,72
132Ø DATA 184,72,192,32,192,32,228,1
2,228,12,252,28,274,126,25Ø,126,25Ø,
126,266,122,25Ø,126,266,13Ø,-1,-1,-1
,-1
133Ø DATA KREBS MIT STERNHAUFEN M44,
-1,-1
134Ø DATA 172,12,172,52,172,76,144,1
ØØ,22Ø,116,-1,-1
135Ø DATA 172,2Ø,172,48,172,6Ø,172,7
2,168,84,156,96,18Ø,84,216,112,184,6
Ø,196,6Ø,196,6Ø,184,72,184,72,184,6Ø
```

```
1360 DATA -1,-1,-1,-1
1370 DATA STIER MIT PLEJADEN,144,104
,44,84,64,24,164,84,224,44,204,124,-
1,-1
1380 DATA 228,32,252,56,280,136,292,
140,196,104,184,104,184,120,168,116,
108,84,60,108,-1,-1
1390 DATA 172,92,72,32,172,92,232,52
,152,112,212,132,212,132,276,140,152
,112,52,92,244,36,244,56,244,56,256,
48
1400 DATA 256,48,244,36,114,118,138,
118,138,118,122,114,138,118,122,122,
152,112,172,92,-1,-1,-1,-1
1410 DATA FUHRMANN,84,84,124,24,184,
44,204,84,184,104,-1,-1
1420 DATA 212,108,120,84,108,84,144,
20,-1,-1
1430 DATA 92,92,132,32,132,32,192,52
,192,52,212,92,212,92,192,112,192,11
2,92,92,190,10,190,34,190,34,186,18
1440 DATA 190,34,194,18,-1,-1,-1,-1
1450 DATA PERSEUS MIT STERNHAUFEN,64
,104,124,64,184,60,224,64,264,64,164
,124,-1,-1
1460 DATA 296,76,276,92,252,108,240,
88,196,116,172,144,152,140,164,84,84
,124,100,76,196,24,212,20,212,36,196
,36
1470 DATA -1,-1
1480 DATA 72,112,132,72,132,72,192,6
8,192,68,232,72,232,72,172,132,232,7
2,272,72,272,72,292,80,210,134,186,1
34
1490 DATA 186,134,202,130,186,134,20
2,138,310,74,310,90,310,90,302,90,30
2,90,310,74,302,94,314,94,314,94,302
,110
1500 DATA 302,110,302,94,-1,-1,-1,-1
1510 DATA SCHWAN,124,24,204,4,144,64
,104,104,44,144,204,124,-1,-1
1520 DATA 204,148,188,92,156,20,156,
36,148,116,148,132,112,96,132,124,84
,84,108,76,108,60,108,44,92,68,92,52
,-1,-1
1530 DATA 52,152,112,112,112,112,152
,72,152,72,132,32,152,72,212,12,152,
72,188,92,188,92,212,132,262,130,234
,130
1540 DATA 234,130,250,126,234,130,25
0,134,90,30,114,30,114,30,102,26,114
,30,102,34,-1,-1,-1,-1
1550 DATA ADLER,144,4,204,4,124,24,1
84,64,84,104,224,124,-1,-1
1560 DATA 228,4,236,60,172,52,124,52
,236,148,140,92,-1,-1
1570 DATA 192,72,212,12,192,72,232,1
32,232,132,236,148,192,72,140,92,140
,92,92,112,192,72,172,52,172,52,152,
12
1580 DATA 152,12,132,32,132,32,124,5
2,124,52,172,52,172,52,132,32,94,34,
118,34,118,34,102,30,118,34,102,38,-
1,-1,-1,-1
1590 DATA ANDROMEDA MIT STERNENNEBEL
 M31,104,4,124,64,144,104,184,144,-1
,-1
1600 DATA 208,72,196,84,164,100,140,
124,128,36,156,52,172,44,-1,-1
1610 DATA 192,152,152,112,152,112,13
2,72,132,72,112,12,132,72,156,52,156
,52,172,44,186,42,186,58,186,58,198,
42
1620 DATA 198,42,186,42,-1,-1,-1,-1
1630 DATA CASSIOPEIA,64,64,104,124,1
64,84,204,104,224,124,264,64,-1,-1
1640 DATA 228,144,172,132,-1,-1
1650 DATA 72,72,112,132,112,132,172,
92,172,92,212,112,212,112,232,132,23
2,132,272,72,-1,-1,-1,-1
1660 DATA PEGASUS,44,104,124,144,204
,104,164,44,124,44,124,4,244,104,280
,84,300,44,-1,-1
1670 DATA 148,84,148,100,188,52,188,
140,228,132,236,108,-1,-1
1680 DATA 212,112,132,52,132,52,172,
52,132,52,132,12,132,52,52,112,52,11
2,132,152,132,152,212,112,212,112,25
2,112
1690 DATA 252,112,288,92,288,92,308,
52,-1,-1,-1,-1
1700 DATA ORION,144,36,152,64,164,92
,224,4,232,144,64,144,84,4,24,64,-1,
-1
1710 DATA 188,68,200,80,212,72,-1,-1
1720 DATA 152,44,160,72,160,72,172,1
00,172,100,240,152,172,100,72,152,72
,152,32,72,32,72,92,12,92,12,152,44
1730 DATA 152,44,232,12,274,14,250,1
4,250,14,266,10,250,14,266,18,30,150
,54,150,54,150,38,146,54,150,38,154
1740 DATA -1,-1,-1,-1
1750 DATA WASSERMANN,204,24,244,44,2
64,124,120,116,-1,-1
1760 DATA 60,60,108,52,100,44,124,28
,156,52,196,20,228,20,228,52,248,68,
196,84,140,108,76,132,68,116,52,116,
-1,-1
1770 DATA 60,60,108,52,108,52,156,52
,156,52,212,32,212,32,228,52,228,52,
252,52,252,52,272,132,252,52,196,84,
196,84
1780 DATA 140,108,140,108,128,124,12
8,124,68,116,68,116,52,116,-1,-1,-1,
```

```
-1
179Ø DATA HERKULES,84,24,144,96,192,
1ØØ,2Ø4,52,244,76,244,52,268,116,2Ø4
,144,-1,-1
18ØØ DATA 128,72,128,84,116,96,172,1
Ø4,148,36,192,44,2Ø4,4Ø,216,88,268,2
Ø,28Ø,4,3Ø4,4,296,16,288,132,28Ø,158
,-1,-1
181Ø DATA 212,6Ø,148,36,212,6Ø,216,8
8,212,6Ø,252,52,252,52,268,2Ø,268,2Ø
,28Ø,4,28Ø,4,3Ø4,4,252,52,252,84,252
,84
182Ø DATA 276,124,276,124,288,132,28
8,132,28Ø,158,252,84,216,88,216,88,2
ØØ,1Ø8,2ØØ,1Ø8,212,152,2ØØ,1Ø8,152,1
Ø4
183Ø DATA 152,1Ø4,116,96,78,34,54,34
,78,34,62,3Ø,78,34,62,38,-1,-1,-1,-1
186Ø REM
188Ø GRAPHICS Ø:SETCOLOR 2,Ø,Ø:SETCO
LOR 4,Ø,5
189Ø POKE 752,1
19ØØ ON STERN GOTO 191Ø,199Ø,2Ø4Ø,2Ø
9Ø,215Ø,222Ø,231Ø,238Ø,255Ø,26ØØ,267
Ø,276Ø,28ØØ,286Ø,293Ø,297Ø
191Ø ? :? "{2SPACES}****{2SPACES}G R
 O S S E R{2SPACES}W A G E N{2SPACES
}****":?
192Ø ? "- Immer zu sehen. (Zirkumpol
ares{8SPACES}Sternbild)":?
193Ø ? "- Die hinteren beiden Sterne
 des{8SPACES}Wagens zeigen zum Polar
stern. Denkt{3SPACES}man sich eine L
inie vom"
194Ø ? "{2SPACES}Polarstern zum Hori
zont findet man{4SPACES}den sogenann
ten Nordpunkt."
195Ø ? "{2SPACES}Alle Sterne, die zw
ischen Polarstern{2SPACES}und Nordpu
nkt liegen sind zu jeder{4SPACES}Zei
t des Jahres sichtbar."
196Ø ? "{2SPACES}Man nennt sie zirku
mpolare Sterne{5SPACES}oder Nordkrei
ssterne.":?
197Ø ? "- Die Sterne im Grossen Wage
n heissen{3SPACES}Dubhe, Merak, Phek
da, Megrez,{9SPACES}Alioth, Mizar un
d Benetnasch."
198Ø GOTO 3Ø6Ø
199Ø ? :? "{2SPACES}******{2SPACES}J
 U N G F R A U{2SPACES}****** ":?
2ØØØ ? "- April bis Juli":? :? "- he
llster Stern heisst Spica":? "{2SPAC
ES}(Spica = Kornaehre)":?
2Ø1Ø ? "- Der Stern rechts oben heis
st":? "{2SPACES}Vindemiatrix (Herrin
des Wein-":? "{2SPACES}gartens). De
r Name stammt aus"
2Ø2Ø ? "{2SPACES}dem Altertum, da de
r Stern kurz{7SPACES}vor{2SPACES}der
 Weinernte am Morgenhimmel{4SPACES}s
ichtbar wird."
2Ø3Ø GOTO 3Ø6Ø
2Ø4Ø ? :? "{2SPACES}******{2SPACES}B
 O O T E S{2SPACES}***** ":?
2Ø5Ø ? "- April bis Oktober":? :? "-
 heller, goldgelb strahlender Stern{
4SPACES}namens Arktur.":?
2Ø6Ø ? "- Doppelstern namens Pulcher
rima{8SPACES}(die Schoenste) nur im
Fernrohr zu{4SPACES}sehen."
2Ø7Ø ? "{2SPACES}Der hellere der bei
den Sterne ist{5SPACES}von goldener,
 der schwaechere von{5SPACES}blasser
, blauer Farbe."
2Ø8Ø GOTO 3Ø6Ø
2Ø9Ø ? :? "{2SPACES}******{2SPACES}L
 O E W E{2SPACES}****** ":?
21ØØ ? "- Maerz bis Mai":? :? "- hel
lster Stern Regulus (Pfeil)":?
211Ø ? "- Denebola ist der letzte zu
m Stern-{4SPACES}bild des Loewen geh
oerende Stern."
212Ø ? "{2SPACES}Der Name kommt aus
dem Arabischen{5SPACES}und bedeutet
'Schwanz des Loewen'.":?
213Ø ? "- Der Loewe gehoert zu den T
ierkreis-{3SPACES}bildern."
214Ø GOTO 3Ø6Ø
215Ø ? :? "{2SPACES}******{2SPACES}K
 R E B S{2SPACES}****** ":?
216Ø ? "- Februar bis Mai":?
217Ø ? "- Die fuenf hellsten Sterne
dieses{6SPACES}Sternbildes, alle von
 4. Groesse,{5SPACES}sind in der For
m eines auf"
218Ø ? "{2SPACES}den Kopf gestellten
 Y angeordnet.":?
219Ø ? "- Sternhaufen M44 (Dreieck)
der dem{5SPACES}blossen Auge nur als
 nebliger Fleck{3SPACES}erscheint."
22ØØ ? "{2SPACES}Der Sternhaufen fue
hrt auch den{7SPACES}Namen Praesepe
(Krippe). Er besteht{3SPACES}aus etw
a 4ØØ Sternen."
221Ø GOTO 3Ø6Ø
222Ø ? :? "{2SPACES}******{2SPACES}S
 T I E R{2SPACES}****** ":?
223Ø ? "- hellster Stern ist Aldebar
an (Pfeil)"
224Ø ? "- Dezember bis April":?
225Ø ? "- Sternengruppe namens Pleja
den"
226Ø ? "{2SPACES}(Dreieck) auch Sieb
```

```
engestirn genannt{2SPACES}Die Namen
der Sterne sind : Alkyone,{2SPACES}A
tlas, Pleione, Asterope,"
2270 ? "{2SPACES}Celaeno, Maja, Elek
tra, Merope und{4SPACES}Taygeta. All
e diese Sterne sind etwa{2SPACES}800
mal groesser als unsere";
2280 ? " Sonne."
2290 ? "{2SPACES}Mit blossem Auge ni
mmt man nur sechs{2SPACES}Sterne des
 Sternenhaufens wahr, der{3SPACES}un
gefaehr 600 Sternen besteht."
2300 GOTO 3060
2310 ? :? "{2SPACES}******{2SPACES}F
 U H R M A N N{2SPACES}****** ":?
2320 ? "- November bis Mai":?
2330 ? "- heller, goldgelber Stern n
amens{7SPACES}Capella (Pfeil). Capel
la ist ein{6SPACES}Riesenstern, dess
en Durch-"
2340 ? "{2SPACES}messer etwa 16mal s
o gross ist wie{4SPACES}der der Sonn
e. Seine Lichtaus-"
2350 ? "{2SPACES}strahlung ist 150ma
l so gross.Er{6SPACES}zeigt eine gel
be Farbe."
2360 ? "{2SPACES}Capella bedeutet 'j
unge Ziege'."
2370 GOTO 3060
2380 ? :? "{2SPACES}******{2SPACES}P
 E R S E U S{2SPACES}****** ":?
2390 ? "- Oktober bis April":?
2400 ? "- Algol (Pfeil) heisst Teufe
l."
2410 ? "{2SPACES}Diesen Namen hat di
eser Stern weil{4SPACES}alle 69 Stun
den seine Helligkeit{6SPACES}fuer et
wa 5 Stunden von der"
2420 ? "{2SPACES}Groesse 2,3 auf die
 Groesse 3,5{7SPACES}herabsinkt. Nac
h fuenf weiteren{7SPACES}Stunden hat
 er dann wieder"
2430 ? "{2SPACES}seine volle Leuchtk
raft erreicht.{5SPACES}Lange Zeit wu
sste man nicht woran{5SPACES}dies la
g."
2440 ? "{2SPACES}Dann aber fand man
heraus, dass{7SPACES}Algol ein Doppe
lstern ist. Einer der{2SPACES}Sterne
 ist ziemlich hell und"
2450 ? "{2SPACES}der andere fast erl
oschen. Da beide{3SPACES}umeinander
kreisen hat man den Ein-{3SPACES}dru
ck, dass die Helligkeit"
2460 ? "{2SPACES}schwankt."
2470 ? "Bitte eine Taste druecken...
"
2480 IF PEEK(764)=255 THEN 2480
2490 POKE 764,255
2500 ? "{CODE125}"
2510 ? :? "{2SPACES}******{2SPACES}P
 E R S E U S{2SPACES}****** ":?
2520 ? "- Zwei Sternenhaufen (Dreiec
ke), die{4SPACES}in klarer Nacht mit
 blossem Auge ge-{2SPACES}sehen werd
en koennen."
2530 ? "{2SPACES}Diese Sternenhaufen
 befinden sich{5SPACES}innerhalb uns
eres Milchstrassen-{6SPACES}systems.
"
2540 GOTO 3060
2550 ? :? "{2SPACES}******{2SPACES}S
 C H W A N{2SPACES}****** ":?
2560 ? "- Mai bis Januar":?
2570 ? "- Doppelstern Albireo (Pfeil
 unten){5SPACES}soll der Kopf des Sc
hwanes sein.":?
2580 ? "- Deneb (Pfeil oben) stellt
den{9SPACES}Schwanz des fliegenden S
chwanes{7SPACES}dar."
2590 GOTO 3060
2600 ? :? "{2SPACES}******{2SPACES}A
 D L E R{2SPACES}****** ":?
2610 ? "- Juni bis Dezember":?
2620 ? "- hellster Stern im Adler is
t Atair.{4SPACES}Sein Licht ist rein
 weiss und er hat{2SPACES}eine 10mal
 hellere Leucht-"
2630 ? "{2SPACES}kraft als unsere So
nne."
2640 ? "{2SPACES}Atair, Wega (Leier)
 und Deneb{9SPACES}(Schwan) gehoeren
 zum Grossen{9SPACES}Sommerdreieck,
dass von Mai"
2650 ? "{2SPACES}bis August die ganz
e Nacht ueber dem{2SPACES}Horizont s
teht und leicht zu finden{3SPACES}is
t.":?
2660 GOTO 3060
2670 ? :? "{2SPACES}******{2SPACES}A
 N D R O M E D A{2SPACES}***** ":?
2680 ? "- Juni bis Maerz":?
2690 ? "- Die Andromeda ist ein ausg
edehntes{4SPACES}Sternbild mit drei
Sternen{12SPACES}3. Groesse.":?
2700 ? "- M31 (Dreieck) ist der gros
se Andro-{3SPACES}medanebel, der in
klaren Naechten{5SPACES}mit blossem
Auge zu sehen ist."
2710 ? "{2SPACES}Nirgendwo kann das
Auge tiefer in{5SPACES}den Weltraum
eindringen als hier,"
2720 ? "{2SPACES}denn der Andromedan
ebel ist{11SPACES}1.5 Millionen Lich
```

```
tjahre entfernt."
2730 ? "{2SPACES}Dieser Nebel ist ei
n aehnliches{7SPACES}milchstrassensy
stem wie unseres und"
2740 ? "{2SPACES}besteht aus Million
en von Sternen."
2750 GOTO 3060
2760 ? :? "{2SPACES}******{2SPACES}C
 A S S I O P E I A{2SPACES}***** ":?

2770 ? "- Immer zu sehen. (Zirkumpol
ares{8SPACES}Sternbild)":?
2780 ? "- Ein schoenes Sternbild in
der Naehe{3SPACES}des Grossen Wagens
. Leicht zu er-{5SPACES}kennen an de
r W-Form."
2790 GOTO 3060
2800 ? :? "{2SPACES}******{2SPACES}P
 E G A S U S{2SPACES}****** ":?
2810 ? "- Juli bis Februar":?
2820 ? "- Beide Seiten des Pegasusvi
erecks{6SPACES}zeigen auf den Polars
tern hin.":?
2830 ? "- In klaren Naechten kann ma
n bis zu{4SPACES}30 Sterne in Innere
n des Vierecks{5SPACES}sehen.":?
2840 ? "- Unterhalb des Pegasus find
et man{6SPACES}die Sterne des Sternb
ildes Wasser-{4SPACES}mann."
2850 GOTO 3060
2860 ? :? "{2SPACES}******{2SPACES}O
 R I O N{2SPACES}***** "
2870 ? :? "- Dezember bis Maerz":? "
- Zwei Sterne 1.Groesse und vier{8SP
ACES}Sterne 2.Groesse"
2880 ? "- Beteigeuze (Pfeil unten) i
st ein{6SPACES}roetlicher Riesenster
n. Sein Durch-{3SPACES}messer ist gr
oesser als die ";
2890 ? "Mars-":? "{2SPACES}bahn."
2900 ? "- Rigel (Pfeil oben)"
2910 ? "- Orionnebel (Dreieck kleine
 Sterne){4SPACES}Riesige Wolke aus l
euchtendem Gas{5SPACES}mit gruenlich
er Farbe."
2920 GOTO 3060
2930 ? :? "{2SPACES}******{2SPACES}W
 A S S E R M A N N{2SPACES}****** ":
?
2940 ? "- Unterhalb des Pegasus befi
ndet sich{3SPACES}Aquarius, der Wass
ermann.":?
2950 ? "- Oestlich des Wassermannes
liegt eine{2SPACES}kleine Gruppe von
 Sternen, die genau{2SPACES}auf dem
Aequator liegt."
2960 GOTO 3060
2970 ? :? "{2SPACES}******{2SPACES}H
 E R K U L E S{2SPACES}****** ":?
2980 ? "- Mai bis November":?
2990 ? "- Der Stern Wega (Pfeil) geh
oert zum{4SPACES}Sternbild Leier und
 fuehrt zum{8SPACES}Herkules.":?
3000 ? "- Im Herkules gibt es den St
ernhaufen{3SPACES}M13, der aus etwa
30000 Sternen be-{3SPACES}steht."
3010 ? "{2SPACES}Seine Entfernung zu
r Erde betraegt{4SPACES}36000 Lichtj
ahre. Es ist ein Kugel-{3SPACES}ster
nhaufen und es lohnt sich"
3020 ? "{2SPACES}ihn mit einem Ferng
las am Himmel zu{3SPACES}suchen."
3030 GOTO 3060
3040 POSITION 2,21:? "AUS DEN STERNE
N GRUESST BERND MOSER":POSITION 2,22
:? "ALLE FREUNDE VON FAMOS SOFTWARE.
..."
3050 FOR WAIT=1 TO 250:NEXT WAIT:RET
URN
3060 ? :? "Bitte druecken Sie eine T
aste ..."
3070 IF PEEK(764)=255 THEN 3070
3080 POKE 764,255
3090 GOTO 600
3100 REM
3140 GRAPHICS 0:SETCOLOR 2,0,0:POKE
752,1
3150 ? "{7SPACES}{CODE160/170/160/20
8/160/204/160/193/160/206/160/197/16
0/212/160/197/160/206/160/170/160}":
? :? "{10SPACES}Waehlen Sie aus:"
3160 ? :? " 1. Sonne{12SPACES}2. Mer
kur{2SPACES}":?
3170 ? " 3. Venus{12SPACES}4. Erde "
:?
3180 ? " 5. Mars{13SPACES}6. Jupiter
":?
3190 ? " 7. Saturn{11SPACES}8. Uranu
s ":?
3200 ? " 9. Neptun{10SPACES}10. Plut
o":?
3210 ? "{9SPACES}11. Sternbilder"
3220 ? :INPUT PLANET:IF PLANET<0 OR
PLANET>11 THEN 3100
3230 PLANET=INT(PLANET)
3240 IF PLANET=0 THEN GOSUB 3040:GOT
O 3100
3250 IF PLANET=11 THEN GOTO 600
3300 GRAPHICS 8:SETCOLOR 2,0,0:SETCO
LOR 0,0,15:COLOR 1
3310 POKE 752,1
3320 RESTORE PLANET*10+3670
3330 READ PLANET$,PLANET2$
3340 ? "{4SPACES}";PLANET$:? "Groess
```

```
envergleich: ERDE und ";PLANET2$
3350 A=60:B=80:IF PLANET2$="SONNE" T
HEN R=0.5:GOTO 3370
3360 A=60:B=80:R=5
3370 GOSUB 3450
3380 READ A,B,R
3390 GOSUB 3450
3400 ? "Weitere Informationen? (J od
er N)"
3410 CLOSE #2:OPEN #2,4,0,"K"
3420 GET #2,A
3430 IF A=ASC("J") THEN 3860
3440 GOTO 3100
3450 PHI=0
3460 Y1=0
3470 X1=R
3480 PHIY=PHI+Y1+Y1+1
3490 PHIXY=PHIY-X1-X1+1
3500 PLOT A+X1,B+Y1
3510 PLOT A-X1,B+Y1
3520 PLOT A+X1,B-Y1
3530 PLOT A-X1,B-Y1
3540 PLOT A+Y1,B+X1
3550 PLOT A-Y1,B+X1
3560 PLOT A+Y1,B-X1
3570 PLOT A-Y1,B-X1
3580 FOR W=1 TO 10:NEXT W
3590 PHI=PHIY
3600 Y1=Y1+1
3610 IF ABS(PHIXY)<ABS(PHIY) THEN PH
I=PHIXY:X1=X1-1
3620 IF X1>=Y1 THEN 3480
3630 RETURN
3680 DATA *****{2SPACES}S O N N E{2S
PACES}**** ,SONNE,160,80,57
3690 DATA *****{2SPACES}M E R K U R{
2SPACES}*** ,MERKUR,160,80,1.8
3700 DATA *****{2SPACES}V E N U S{2S
PACES}**** ,VENUS,160,80,4.7
3710 DATA *****{2SPACES}E R D E{2SPA
CES}**** ,MOND,160,80,1.3
3720 DATA *****{2SPACES}M A R S{2SPA
CES}**** ,MARS,160,80,2.6
3730 DATA *****{2SPACES}J U P I T E
R{2SPACES}**** ,JUPITER,160,80,57.4
3740 DATA *****{2SPACES}S A T U R N{
2SPACES}**** ,SATURN,160,80,47.4
3750 DATA *****{2SPACES}U R A N U S{
2SPACES}**** ,URANUS,160,80,19.8
3760 DATA *****{2SPACES}N E P T U N{
2SPACES}**** ,NEPTUN,160,80,17.5
3770 DATA *****{2SPACES}P L U T O{2S
PACES}**** ,PLUTO,160,80,2.3
3780 ? :? "Bitte druecken Sie eine T
aste ..."
3790 IF PEEK(764)=255 THEN 3790
3800 POKE 764,255
3810 GOTO 3100
3860 GRAPHICS 0:SETCOLOR 2,0,0
3870 ON PLANET GOTO 3880,3920,3980,4
030,4090,4150,4210,4270,4330,4390
3880 ? :? "{8SPACES}***{2SPACES}S O
N N E{2SPACES}***"
3890 ? :? "-{2SPACES}DURCHMESSER IN
KM : 1.391.000"
3900 ? :? "-{2SPACES}ROTATIONSDAUER{
4SPACES}: 25-27 TAGE"
3910 GOTO 3780
3920 ? :? "{8SPACES}***{2SPACES}M E
R K U R{2SPACES}***"
3930 ? :? "-{2SPACES}DURCHMESSER IN
KM : 4.700"
3940 ? :? "-{2SPACES}ROTATIONSDAUER{
4SPACES}: 88 TAGE"
3950 ? :? "-{2SPACES}ENTF. ZUR SONNE
{3SPACES}: 58 MILL. KM"
3960 ? :? "-{2SPACES}UMLAUFSZEIT{7SP
ACES}: 88 TAGE"
3970 GOTO 3780
3980 ? :? "{8SPACES}***{2SPACES}V E
N U S{2SPACES}***"
3990 ? :? "-{2SPACES}DURCHMESSER IN
KM : 12.200"
4000 ? :? "-{2SPACES}ENTF. ZUR SONNE
{3SPACES}: 108 MILL. KM"
4010 ? :? "-{2SPACES}UMLAUFSZEIT{7SP
ACES}: 225 TAGE"
4020 GOTO 3780
4030 ? :? "{8SPACES}***{2SPACES}E R
D E{2SPACES}***"
4040 ? :? "-{2SPACES}DURCHMESSER IN
KM : 12.756"
4050 ? :? "-{2SPACES}ROTATIONSDAUER{
4SPACES}: 23h 56m 4s"
4060 ? :? "-{2SPACES}ENTF. ZUR SONNE
{3SPACES}: 149 MILL. KM"
4070 ? :? "-{2SPACES}UMLAUFSZEIT{7SP
ACES}: 1 JAHR"
4080 GOTO 3780
4090 ? :? "{8SPACES}***{2SPACES}M A
R S{2SPACES}***"
4100 ? :? "-{2SPACES}DURCHMESSER IN
KM : 6.770"
4110 ? :? "-{2SPACES}ROTATIONSDAUER{
4SPACES}: 24h 37m 23s"
4120 ? :? "-{2SPACES}ENTF. ZUR SONNE
{3SPACES}: 226 MILL. KM"
4130 ? :? "-{2SPACES}UMLAUFSZEIT{7SP
ACES}: 1,88 JAHRE"
4140 GOTO 3780
4150 ? :? "{8SPACES}***{2SPACES}J U
P I T E R{2SPACES}***"
4160 ? :? "-{2SPACES}DURCHMESSER IN
KM : 146.500"
```

```
4170 ? :? "-{2SPACES}ROTATIONSDAUER{
4SPACES}: 10h"
4180 ? :? "-{2SPACES}ENTF. ZUR SONNE
{3SPACES}: 778 MILL. KM"
4190 ? :? "-{2SPACES}UMLAUFSZEIT{7SP
ACES}: 11,86 JAHRE"
4200 GOTO 3780
4210 ? :? "{8SPACES}***{2SPACES}S A
T U R N ***"
4220 ? :? "-{2SPACES}DURCHMESSER IN
KM : 121.000"
4230 ? :? "-{2SPACES}ROTATIONSDAUER{
4SPACES}: 10,5h"
4240 ? :? "-{2SPACES}ENTF. ZUR SONNE
{3SPACES}: 1428 MILL. KM"
4250 ? :? "-{2SPACES}UMLAUFSZEIT{7SP
ACES}: 29,46 JAHRE"
4260 GOTO 3780
4270 ? :? "{8SPACES}***{2SPACES}U R
A N U S{2SPACES}***"
4280 ? :? "-{2SPACES}DURCHMESSER IN
KM : 50.700"
4290 ? :? "-{2SPACES}ROTATIONSDAUER{
4SPACES}: 10,75h"
4300 ? :? "-{2SPACES}ENTF. ZUR SONNE
{3SPACES}: 2868 MILL. KM"
4310 ? :? "-{2SPACES}UMLAUFSZEIT{7SP
ACES}: 84,02 JAHRE"
4320 GOTO 3780
4330 ? :? "{8SPACES}***{2SPACES}N E
P T U N{2SPACES}***"
4340 ? :? "-{2SPACES}DURCHMESSER IN
KM : 44.800"
4350 ? :? "-{2SPACES}ROTATIONSDAUER{
4SPACES}: 15,75h"
4360 ? :? "-{2SPACES}ENTF. ZUR SONNE
{3SPACES}: 4494 MILL. KM"
4370 ? :? "-{2SPACES}UMLAUFSZEIT{7SP
ACES}: 164,80 JAHRE"
4380 GOTO 3780
4390 ? :? "{8SPACES}***{2SPACES}P L
U T O{2SPACES}***"
4400 ? :? "-{2SPACES}DURCHMESSER IN
KM : 6.000"
4410 ? :? "-{2SPACES}ENTF. ZUR SONNE
{3SPACES}: 5917 MILL. KM"
4420 ? :? "-{2SPACES}UMLAUFSZEIT{7SP
ACES}: 248,60 JAHRE"
4430 GOTO 3780
  READY !
```

Princess

Beschreibung des Programmes (Atari 600XL/800XL, 9300 Bytes)

Befreien Sie die Prinzessin aus den Händen von schießwütigen Wesen. Vorsicht! Einige können durch Hüpfen schlagartig ihren Ort wechseln. Um die Prinzessin erlösen zu können, müssen Räume verschiedenster Art erfolgreich durchquert werden. Zu Beginn des Spieles hat der Spieler 300 Lebenseinheiten, mit denen er auskommen muß. Jeder Treffer der Wesen verbraucht solche Einheiten. Die Lebenseinheiten stehen rechts oben am Bildschirm. Welchen Ausgang man lebend erreichen muß, steht zu Beginn auf dem Bildschirm links unten. Ein Pfeil erinnert an die Richtung, in der der Ausgang liegt. Es gibt vier Ausgänge, in jeder Himmelsrichtung einen. Bevor man jedoch durch den Ausgang den nächsten Raum erreichen kann, muß man die Wache durch zwei Schüsse außer Gefecht setzen und deren Waffe an sich nehmen. Die Wache ist der invers dargestellte Pfeil, der nach zwei Treffern normal aussieht.

Man darf sich jedoch nicht von der Wache treffen lassen. Um die Waffe an sich zu bringen, muß man mit einem Stern(*=Spieler) auf den Pfeil auflaufen. Jetzt kann man durch den Ausgang in den zweiten Raum eindringen. Im zweiten Raum befindet sich im Zentrum des Bildschirmes ein Tresor, der aufgeschossen werden muß um den dort befindlichen Schlüssel zu holen (auch durch auflaufen). In dieser Runde des Spiels wird ihr nächster Ausgang neu festgelegt. Gehen Sie nun dorthin, dann kommen Sie in den dritten Raum. Versuchen Sie zu überleben und gehen Sie zum nächsten Ausgang. In der vierten Runde müssen Sie erneut den Ausgang möglichst unbeschadet erreichen. Nun sind Sie endlich im fünften

Raum, im Zentrum der Festung. In der Mitte des Bildschirmes befindet sich jetzt das Gefängnis, in dem die Prinzessin gefangengehalten wird. - Sie haben zwar den Schlüssel, aber es gibt trotzdem noch Schwierigkeiten, da sich nur zeitweise die Gelegenheit bietet, hinein und heraus zu kommen. Die Prinzessin befreien Sie, in dem Sie auflaufen. Nun müssen Sie mit ihr den Ausgang erreichen.

Bei diesem Spiel kann man höchstens drei Bewegungen machen, drei Schüsse abgeben oder eine Kombination daraus - dann sind die Wesen dran. Erst wenn alle Wesen geschossen haben, ist der Spieler wieder an der Reihe. Beim Schießen kann man zielen und den Schuß lenken, indem man auf den Feuerknopf drückt und den Steuerknüppel in die gewünschte Richtung lenkt. Mit dem Drücken des Feuerknopfes beginnt auch die jeweilige Runde. Wolfgang Schmidt

```
1Ø REM * PRINCESS *
11 REM * FUER CHIP-SPECIAL*
12 REM * VON WOLFGANG SCHMIDT*
1ØØ GOSUB 5ØØØ:REM ** PRINCESS **
11Ø DIM J(15),V(3Ø),S$(6),R$(4):R$="
\{CODE95}]^":IF C=Ø THEN 9ØØ
115 FOR I=5 TO 15:READ D:J(I)=D:NEXT
 I:J=632:L=15:POKE 752,1:LF=3ØØ:R=53
77Ø:S=96:WF=1:KY=1
12Ø SC=PEEK(88)+256*PEEK(89)
13Ø S$="{CODE18/18/18}{3SPACES}"
14Ø C=INT(RND(Ø)*88Ø)+SC+4Ø
2ØØ COLOR 16Ø:PLOT Ø,Ø:DRAWTO 39,Ø:D
RAWTO 39,23:DRAWTO Ø,23:DRAWTO Ø,Ø
21Ø GOSUB 2ØØØ
22Ø POKE C,1Ø
25Ø FOR I=1 TO L
26Ø V(I)=INT(RND(Ø)*88Ø)+SC+4Ø:IF PE
EK(V(I))>Ø THEN 26Ø
27Ø POKE V(I),96-(FD=2)*64:NEXT I
28Ø G=Ø:GOTO 4ØØ
3ØØ REM * SOLDATEN:SCHIESSEN+BEWEGEN
 *
3Ø2 N=C:GOSUB 12ØØ:Q1=X:Q2=Y:SH=111:
POKE 77,Ø
3Ø5 FOR Z=1 TO L:IF V(Z)=SC THEN 39Ø
31Ø V=V(Z):N=V:GOSUB 12ØØ
32Ø B=V+4Ø*((Q2>Y)-(Q2<Y))+(Q1>X)-(Q
1<X)
325 GOSUB 35ØØ
33Ø IF S=96 THEN 34Ø
335 B=INT(RND(Ø)*88Ø)+SC+2Ø:IF PEEK(
B)>Ø THEN 335:REM * TELEPORTIEREN *
34Ø IF PEEK(B)=Ø THEN POKE V,Ø:POKE
B,S:V(Z)=B
35Ø IF B=C THEN 7ØØ
36Ø SOUND Ø,(ABS(C-V))/4+15,1Ø,8
39Ø NEXT Z:SOUND Ø,Ø,Ø,Ø:G=Ø:IF LF<=
Ø THEN 7ØØ
392 IF FD+1<>WF THEN 4ØØ
394 N=WP:GOSUB 12ØØ:B=WP+4Ø*((Q2>Y)-
(Q2<Y))+(Q1>X)-(Q1<X):SH=212:V=WP:GO
SUB 35ØØ:POKE WP,WA
396 WP=WP+4Ø*((YA>Y)-(YA<Y))+(XA>X)-
(XA<X):WA=PEEK(WP):IF WA=OUT+144 THE
N WF=WF+1:GOTO 4ØØ
398 POKE WP,22Ø
4ØØ G=G+1:IF G>3 THEN 3ØØ:REM * SPIE
LER *
4Ø1 GOSUB 4ØØØ
4Ø2 IF FD>3 THEN POSITION 19,9:H=(PE
EK(19)/2)=(INT(PEEK(19)/2)):? S$(1+H
*3*KY,3+H*3*KY);
4Ø3 IF PEEK(J)=15 THEN 4Ø2
4Ø5 E=PEEK(J):IF PEEK(644)=Ø THEN GO
SUB 3ØØØ:GOTO 4ØØ
4Ø6 POKE C,Ø:C=C+J(E):T=PEEK(C):IF L
F<=Ø THEN 7ØØ
4Ø7 IF T=82 THEN 43Ø
41Ø IF T=Ø THEN POKE C,1Ø:FOR KK=1 T
O 1ØØ:NEXT KK:GOTO 4ØØ
42Ø IF T=OUT+144 THEN 95Ø
425 IF T=92 THEN FOR I=255 TO 12Ø ST
EP -2:SOUND Ø,I,14,1Ø:NEXT I:SOUND Ø
,Ø,Ø,Ø:POKE C,1Ø:GOTO 4ØØ
43Ø IF T=82 THEN LF=LF-3Ø:C=INT(RND(
8)*84Ø)+2Ø+SC:POKE C,1Ø:GOTO 3ØØ
435 IF T=11 THEN KY=1:FOR I=1 TO 1ØØ
:SOUND Ø,I,1Ø,1Ø:NEXT I:SOUND Ø,Ø,Ø,
Ø:POKE C,1Ø:GOTO 4ØØ
44Ø LF=LF-5:GOSUB 4ØØØ
45Ø E=PEEK(J):POKE C,8Ø:FOR W=1 TO 1
ØØ:SOUND Ø,W,2,8:NEXT W:SOUND Ø,Ø,Ø,
Ø:POKE C,T:C=C-J(E):POKE C,1Ø
46Ø GOTO 4ØØ
7ØØ ? "+{CODE16Ø/21Ø/213/2ØØ/197/16Ø
/211/193/2Ø6/198/212}+"
75Ø FOR I=1 TO 3:FOR S=1ØØ TO 2ØØ:SO
UND Ø,S,1Ø,1Ø:SOUND Ø,S+2,1Ø,1Ø:NEXT
 S:NEXT I
8ØØ SOUND Ø,Ø,Ø,Ø:GOSUB 15ØØ
8Ø2 POSITION Ø,23:PRINT "{2SPACES}{C
ODE21Ø/2Ø7/212/197/21Ø/16Ø/2Ø3/2Ø6/2
Ø7/2Ø8/198/16Ø/196/21Ø/197/213/195/2
```

```
Ø3/197/2Ø6/16Ø/189/16Ø/2Ø6/197/213/1
97/211/16Ø/211/2Ø8/2Ø1/197/2Ø4}{2SPA
CES}";
8Ø5 IF STICK(Ø)+PEEK(644)=16 THEN 8Ø
5
81Ø IF PEEK(644)=Ø THEN RUN
9ØØ POKE 752,Ø:POKE 82,2:END
92Ø DATA 41,-39,1,Ø,39,-41,-1,Ø,4Ø,-
4Ø,Ø
95Ø REM *OUT*
96Ø FD=FD+1:IF FD>4 THEN 985
97Ø ? CHR$(125):C=C+84Ø*((C-SC<17Ø)-
(C-SC>8ØØ))+37*((Q1<4)-(Q1>35))
98Ø GOTO 2ØØ
985 IF FD=5 THEN OUT=Ø:RESTORE :FOR
I=1 TO 8:READ D:POKE C+D,64:NEXT I:P
OKE C-4Ø,Ø:GOTO 2ØØ
99Ø FOR I=SC TO SC+959:POKE I,64:NEX
T I:POSITION 8,12:? "{CODE196/213/16
Ø/2ØØ/193/211/212/16Ø/197/211/16Ø/19
9/197/211/195/2ØØ/193/198/198/212/16
Ø/161/161/161}":FOR I=23Ø TO 1ØØ STE
P -1:SOUND Ø,I,1Ø,1Ø
995 SOUND 1,I-2,1Ø,1Ø:SOUND 2,I+2,1Ø
,1Ø:NEXT I:RESTORE 1ØØØ:SOUND 2,Ø,Ø,
Ø
1ØØØ REM * SIEGER-HYMNE *
1ØØ5 READ T,L:IF T<Ø THEN 9ØØ
1Ø1Ø SOUND Ø,T,1Ø,1Ø:SOUND 1,T,1Ø,1Ø
:SOUND 2,T,1Ø,1Ø
1Ø2Ø FOR W=1 TO 2ØØ/L:NEXT W
1Ø3Ø SOUND Ø,Ø,Ø,Ø:GOTO 1ØØØ
1Ø5Ø DATA 121,2,91,2.67,91,8,91,1
1Ø6Ø DATA 121,2,81,2.67,96,8,91,1
1Ø7Ø DATA 121,2,91,2.67,72,8,6Ø,2,72
,2.67
1Ø8Ø DATA 91,8,91,2.67,96,2.67,91,8,
81,1
1Ø9Ø DATA 121,2,91,2.67,91,8,91,1,12
1,2.67,81,2.67,96,8,91,1
11ØØ DATA 121,2,91,2.67,72,8,6Ø,2,72
,2
111Ø DATA 91,8,121,2,81,2.67,96,8,91
,1
118Ø DATA -1,-1
12ØØ REM * POKE-XY-WANDLER *
121Ø P=N-SC:Y=INT(P/4Ø):X=P-Y*4Ø:RET
URN
15ØØ RESTORE 18ØØ:FOR W=1 TO 3ØØ:NEX
T W
151Ø READ T,L:IF T<Ø THEN RETURN
152Ø SOUND Ø,T,1Ø,1Ø:FOR W=1 TO L*2Ø
:NEXT W:SOUND Ø,Ø,Ø,Ø:GOTO 151Ø
18ØØ DATA 162,8,121,1Ø,162,8,121,2,9
6,1Ø,121,8,96,6,81,Ø8,121,2,162,8,Ø,
1,162,8,121,12,-1,-1
2ØØØ REM * HINDERNISSE *
2Ø1Ø FOR I=1 TO 5:POKE SC+17+I,144:P
OKE SC+937+I,146
2Ø15 POKE SC+I*4Ø+36Ø,147:POKE SC+I*
4Ø+399,145:NEXT I
2Ø2Ø HN=INT(RND(2)*4)+1
2Ø3Ø OUT=INT(RND(Ø)*4):IF WF=FD+1 TH
EN 25ØØ
2Ø4Ø POSITION 2,23:? "AUSGANG: ";OUT
;:POKE SC,ASC(R$(OUT+1,OUT+1))
2Ø5Ø IF PEEK(644)=1 AND FD<>4 THEN 2
Ø5Ø
2Ø6Ø POSITION 2,23:? "{CODE16Ø/16Ø/1
6Ø/16Ø/16Ø/16Ø/16Ø/16Ø/16Ø/16Ø/16Ø/1
6Ø}";
2Ø7Ø GOSUB 4ØØØ:ON FD+1 GOTO 2Ø9Ø,25
5Ø,26ØØ,263Ø,24ØØ,27ØØ
2Ø9Ø ON HN GOTO 21ØØ,221Ø,222Ø,23ØØ
21ØØ PLOT 4,6:DRAWTO 4,2:DRAWTO 1Ø,2
211Ø PLOT 18,4:DRAWTO 22,4:PLOT 5,11
:DRAWTO 9,11:PLOT 7,13:DRAWTO 7,9
212Ø PLOT 29,2:DRAWTO 35,2:DRAWTO 35
,6
213Ø PLOT 34,11:DRAWTO 3Ø,11:PLOT 32
,13:DRAWTO 32,9
214Ø PLOT 18,19:DRAWTO 22,19:PLOT 29
,21:DRAWTO 35,21:DRAWTO 35,17
215Ø PLOT 4,17:DRAWTO 4,21:DRAWTO 1Ø
,21
218Ø RETURN
221Ø FOR I=1 TO PEEK(R):P=INT(RND(Ø)
*88Ø)+SC:IF PEEK(P)=Ø THEN POKE P,12
8:SOUND Ø,PEEK(R),4,8
2215 NEXT I:SOUND Ø,Ø,Ø,Ø:RETURN
222Ø FOR I=5 TO 18 STEP 2:PLOT 1Ø,I:
PLOT 11,I+1:PLOT 9,I+1:PLOT 3Ø,I:PLO
T 31,I+1:PLOT 29,I+1:NEXT I
225Ø RETURN
23ØØ PLOT 16,4:DRAWTO 2Ø,8:DRAWTO 24
,4:PLOT 16,19:DRAWTO 2Ø,15:DRAWTO 24
,19
231Ø PLOT 4,8:DRAWTO 8,12:DRAWTO 4,1
6:PLOT 35,8:DRAWTO 31,12:DRAWTO 36,1
7:RETURN
24ØØ POSITION 2Ø,12:? "{CODE2Ø8}":PL
OT 19,8:DRAWTO 18,8:DRAWTO 18,14:DRA
WTO 22,14:DRAWTO 22,8:DRAWTO 21,8
241Ø PLOT 19,1Ø:PLOT 21,1Ø
2415 OUT=32:POSITION 2,23:? "Rette s
ie !";:S=96:FOR W=1 TO 3ØØ:NEXT W:PO
SITION 2,23:? "{CODE16Ø/16Ø/16Ø/16Ø/
16Ø/16Ø/16Ø/16Ø/16Ø/16Ø}";
245Ø RETURN
25ØØ REM 1.QDT
251Ø WP=INT(RND(Ø)*88Ø)+SC+4Ø:IF PEE
K(WP)>Ø THEN 251Ø
252Ø POKE WP,22Ø
253Ø YA=12*(OUT=1 OR OUT=3)+23*(OUT=
```

```
2):XA=20*(OUT=0 OR OUT=2)+39*(OUT=1)
2540 GOTO 2040
2550 COLOR 35:PLOT 16,8:DRAWTO 18,10
:DRAWTO 18,14:DRAWTO 16,16:PLOT 16,1
2:PLOT 17,12
2560 PLOT 18,10:DRAWTO 22,10:DRAWTO
24,8:PLOT 22,10:DRAWTO 22,14:DRAWTO
24,16
2570 PLOT 22,14:DRAWTO 18,14:PLOT 20
,8:PLOT 20,9:PLOT 23,12:PLOT 24,12:P
LOT 20,15:PLOT 20,16:COLOR 43:PLOT 2
0,12:COLOR 160:GOTO 2090
2600 S=32:GOTO 2090
2630 S=96:GOTO 2090
2650 OUT=32:POSITION 2,23:? "Rette s
ie !";:S=96:FOR W=1 TO 300:NEXT W:PO
SITION 2,23:? "{CODE160/160/160/160/
160/160/160/160/160/160/160}";:GOTO
2090
2700 RETURN
3000 REM *SHT*
3100 K=0:SH=84
3110 K=K+1:IF PEEK(C+K*J(E))=0 THEN
POKE C+K*J(E),SH:SOUND 0,K*4,8,8:GOT
O 3110
3120 U=C+K*J(E):T=PEEK(U):FOR I=C+J(
E) TO K*J(E)+C STEP J(E)
3130 POKE I,0:NEXT I
3140 FOR I=1 TO 50:POKE U,PEEK(R):SO
UND 0,PEEK(R),12,8:NEXT I:SOUND 0,0,
0,0
3150 FOR I=1 TO L:IF V(I)=U THEN V(I
)=SC:POKE U,0:RETURN
3160 NEXT I:IF T<>S THEN POKE U,T:IF
 T=3 AND WF=8 THEN POKE U,0
3165 IF U=WP THEN TF=TF+1:IF TF>1 TH
EN POKE U,92:WF=8:WP=SC
3170 RETURN
3500 REM #SCHUSS#
3600 A=B-V:K=0
3610 K=K+1:IF PEEK(V+K*A)=0 THEN POK
E V+K*A,SH:SOUND 0,K*5,20,8:GOTO 361
0
3620 U=V+K*A:T=PEEK(U):FOR I=V+A TO
V+K*A STEP A
3630 POKE I,0:NEXT I
3640 FOR I=1 TO L:POKE U,PEEK(R)
3650 IF V(I)=U THEN V(I)=SC
3680 SOUND 0,PEEK(R),14,8:NEXT I:POK
E U,0
3690 SOUND 0,0,0,0:IF T<>96 THEN POK
E U,T
3695 IF U=C THEN LF=LF-INT(RND(0)*40
):POKE U,10:GOSUB 4000:IF V=WP THEN
3710
3700 RETURN
3710 FOR I=L TO L+5
3720 V(I)=INT(RND(0)*880)+SC+40:IF P
EEK(V(I))>0 THEN 3720
3730 POKE V(I),96:NEXT I:L=L+5:RETUR
N
4000 POSITION 30,0:PRINT "LIFE:";LF;
"{CODE160}":RETURN
5000 REM * VORSPANN *
5100 POKE 82,0:GRAPHICS 0:PRINT "{CO
DE170/170/170/170/170/170/170/170/17
0/170/160/208/160/210/160/201/160/20
6/160/195/160/197/160/211/160/211/16
0/170/170/170/170/170/170/170/170/17
0/170/170/170/170}"
5120 PRINT "{CODE29}MITWIRKENDE:"
5130 PRINT "{CODE29}* - SIE"
5140 PRINT "{CODE96} - SOLDATEN"
5150 PRINT "{CODE27/156} - WAECHTER"
5160 PRINT "{CODE64} - TELEPORTER"
5200 PRINT "{CODE29/29}IHR AUFTRAG:"
5210 PRINT "{CODE29}1. RAUBEN SIE DI
E WAFFE DES WAECHTERS({CODE27/28})";
5220 PRINT "2. NEHMEN SIE IM 2.FELD
DEN SCHLUESSEL:+";:PRINT "3. BEFREIE
N SIE IN FELD#5 DIE PRINZESSIN"
5300 PRINT "{CODE29/29}NEHMEN SIE DE
N AUFTRAG AN ?":PRINT "JA{3SPACES}=
ROTEN KNOPF DRUECKEN"
5310 PRINT "NEIN = STEUERKNUEPPEL BE
WEGEN"
5320 IF STICK(0)+PEEK(644)=16 THEN 5
320
5340 C=PEEK(644)=0:PRINT "{CODE125}"
:RETURN

 READY !
```

Nuclear

Beschreibung des Programmes (Atari 600XL/800XL, 4720 Bytes)
Nuclear ist ein Strategiespiel mit 10x10 Feldern. Es geht darum, auf möglichst vielen Feldern eine fast kritische Masse aufzubauen. Wird nun auf einem Feld die kritische Masse erreicht, explodiert der "Atomkern", die Bruchstücke verteilen sich symmetrisch auf die direkt angrenzenden vertikalen und horizontalen Felder. Dort wird nun ebenfalls kritische Masse erreicht – es kann zur Kettenreaktion kommen. Eine kritische Masse ist erreicht, wenn auf einem Spielfeld so viele Spielsteine stehen, wie direkt angrenzende vertikale und horizontale Felder vorhanden sind:

```
  A B C D E F . .
1 2 3 3 3 3 3 . .
2 3 4 4 4 4 4 . .
3 3 4 4 4 4 4 . .
4 3 4 4 4 4 4 . .
5 3 4 4 4 4 4 . .
. . . . . . . . .
. . . . . . . . .
```

Die Zahlen in den Feldern geben die kritische Masse an, also die Anzahl der Spielsteine die zur "Explosion" führen.
Hat mein Gegenspieler eine fast kritische Masse aufgebaut und diese wird von einem meiner Bruchstücke getroffen, so wandeln sich alle Spielsteine in die meinen um. Durch eine gezielte Kettenreaktion kann ich alle Steine meines Gegners in die meinen umwandeln und habe gewonnen. Durch die Eingabe der x-y-Werte des Rasters wird der Ort des neuen Steines definiert.

Wolfgang Schmidt

```
100 REM * NUCLEAR *
101 REM * FUER CHIP-SPECIAL*
102 REM * VON WOLFGANG SCHMIDT*
170 OPEN #1,4,Ø,"K":IF PEEK(87)>Ø TH
EN GRAPHICS Ø
180 DIM V(11,11),F(11,11),P$(3Ø),N$(
15)
190 GRAPHICS Ø:POKE 82,2:P=Ø:PRINT "
{CODE125}*** NUCLEAR ***":WAIT=15ØØ
200 REM *** SPIELREGELN ***
210 PRINT "{CODE29}Sind die Spielreg
eln bekannt ?";:GET #1,E:PRINT CHR$(
E):IF E<>78 THEN 3ØØ
215 PRINT "{CODE29}*** SPIELREGELN *
**"
220 PRINT "Auf einem 1Øx1Ø Felder gr
ossen Gebiet wird von jedem Spieler
abwechselnd{4SPACES}eine Ladung auf
ein freies"
230 PRINT "oder von eigenen Ladungen
 besetztes{3SPACES}Feld gesetzt.{2SP
ACES}Erreicht die Gesamt-{3SPACES}la
dung eine kritische Masse,"
235 PRINT "(=Anz. d. direkt angrenze
nden Felder),";
240 PRINT "so explodiert diese und v
erteilt sich auf die max. 4 umliegen
den Felder."
250 PRINT "Falls diese bereits Ladun
gen tragen,{2SPACES}wird zu dieser 1
 addiert - ist damit{2SPACES}eine kr
itische Masse erreicht,"
260 PRINT "so explodiert diese - die
 Ketten-{5SPACES}reaktion startet."
270 PRINT "Der Spieler gewinnt, dess
en Ladungen{2SPACES}als einzige auf
dem Spielfeld uebrig{2SPACES}bleiben
."
280 GOTO 31Ø
290 PRINT "{CODE29}FERTIG ?";:GET #1
,E:GOTO 4ØØ
300 REM *** INIT DER VARIABLEN ***
305 PRINT "{CODE29}MOMENT BITTE !"
310 FOR X=Ø TO 11
320 FOR Y=Ø TO 11
330 V(X,Y)=Ø:F(X,Y)=1Ø:NEXT Y:NEXT X
340 FOR X=1 TO 1Ø:FOR Y=1 TO 1Ø:F(X,
Y)=Ø
350 V(X,Y)=4-(X<2 OR X>9 OR Y<2 OR Y
>9)
360 NEXT Y:NEXT X
370 V(1,1)=2:V(1Ø,1Ø)=2:V(1,1Ø)=2:V(
1Ø,1)=2
380 IF E=78 THEN 29Ø
400 REM *** SPIELFELD AUFBAU ***
410 GRAPHICS 1:POKE 71Ø,Ø:FOR X=1 TO
 1Ø
420 POSITION X,Ø:PRINT #6;CHR$(64+X)
```

```
:POSITION Ø,X:PRINT #6;X-1:POSITION
X,11:PRINT #6;CHR$(64+X):POSITION 11
,X:PRINT #6;X-1
425 FOR Y=1 TO 1Ø
43Ø POSITION X,Y:PRINT #6;"."
44Ø NEXT Y:NEXT X
442 POSITION Ø,14:PRINT #6;"EXPLOSIO
NEN:"
45Ø PRINT "Name des 1. Spielers: ";:
INPUT N$
46Ø L1=LEN(N$):P$(1,L1)=N$
47Ø PRINT "Name des 2. Spielers: ";:
INPUT N$
48Ø P$(L1+1)=N$:L2=LEN(N$)
5ØØ REM *** SPIEL ***
51Ø Z=Z+1
52Ø PRINT CHR$(125);P$(P*L1+1,P*L2+L
1);"'s Zug#";Z;":";
53Ø GET #1,H:PRINT CHR$(H);:W=1
54Ø IF H<65 OR H>74 THEN PRINT :PRIN
T "<<< WOHIN ??? >>>":GOSUB WAIT:GOT
O 52Ø
55Ø GET #1,V:PRINT CHR$(V)
56Ø IF V<48 OR V>57 THEN PRINT "DIE
LADUNG LANDETE WOHL IN DEINEM HIRN":
GOSUB WAIT:GOTO 52Ø
57Ø H=H-64:V=V-47:EX=Ø
58Ø R=(P=1)-(P=Ø)
6ØØ REM * PLATZ-VERIFY *
61Ø IF SGN(F(H,V))=-R THEN PRINT "<<
< FEINDLICHES GEBIET >>>":GOSUB WAIT
:GOTO 52Ø
62Ø F(H,V)=F(H,V)+R
63Ø POSITION H,V:FOR I=6Ø TO 1Ø STEP
 -2:SOUND Ø,I,1Ø,1Ø:NEXT I:SOUND Ø,Ø
,Ø,Ø
635 PRINT #6;CHR$(ABS(F(H,V))+128*P+
16)
64Ø IF ABS(F(H,V))<V(H,V) THEN GOTO
92Ø
7ØØ X=H:Y=V:E=Ø:GOSUB 2ØØØ:Q1=Ø
71Ø IF E=Ø THEN 8ØØ
72Ø E=Ø:FOR X=1 TO 1Ø
73Ø FOR Y=1 TO 1Ø
74Ø IF ABS(F(X,Y))>=V(X,Y) THEN GOSU
B 2ØØØ
75Ø NEXT Y:NEXT X
76Ø IF E=1 AND Q1<99 THEN 72Ø
77Ø W=1:IF Q1>=99 THEN W=-1
8ØØ REM *EXISTENCE-TEST*
81Ø FOR X=1 TO 1Ø
82Ø FOR Y=1 TO 1Ø
83Ø IF SGN(F(X,Y))=-R THEN Q1=Ø:GOTO
 92Ø-2ØØ*(W<Ø)
9ØØ NEXT Y:NEXT X
91Ø POSITION Ø,14:PRINT #6;"SPIELER
";P$(P*L1+1,P*L2+L1);"{4SPACES}":PRI
NT #6;"HAT GEWONNEN !"
912 FOR Q1=1 TO 3
913 FOR I=22Ø-Q1*1Ø TO 3Ø-Q1*1Ø STEP
 -2Ø
914 FOR E=I TO I+1Ø STEP 2:SOUND Ø,E
,1Ø,5+E-I:NEXT E:NEXT I:NEXT Q1
915 FOR I=1Ø TO 25Ø STEP 2:SOUND Ø,I
,Ø,15:POKE 7Ø8,I:NEXT I
918 SOUND Ø,Ø,Ø,Ø:PRINT "{CODE125}WO
LLT IHR NOCHMAL SPIELEN ?";:GET #1,I
:IF I=74 OR I=89 THEN RUN
919 GRAPHICS Ø:END
92Ø REM
99Ø P=P-R:W=1:GOTO 51Ø+P*1Ø
15ØØ FOR I=1 TO 5ØØ:NEXT I:RETURN
2ØØØ REM *** EXPLODE-ROUTINE ***
212Ø RESTORE :FOR I=1 TO 4
213Ø READ DX,DY:SOUND Ø,4Ø-I*1Ø,Ø,1Ø
214Ø L=F(X+DX,Y+DY):IF L>9 THEN 22ØØ
215Ø F(X+DX,Y+DY)=ABS(L)*R+R
2155 REM CØ=CØ-L*R:C1=C1+L*R:POSITIO
N 2,15:PRINT #6;CØ:POSITION 6,15:PRI
NT #6;C1
216Ø IF ABS(F(X+DX,Y+DY))>=V(X+DX,Y+
DY) THEN E=1
217Ø POSITION X+DX,Y+DY
218Ø PRINT #6;CHR$(ABS(F(X+DX,Y+DY))
+128*P+16):F(X,Y)=F(X,Y)-R
2185 POSITION X,Y:PRINT #6;"."
219Ø IF F(X,Y)<>Ø THEN POSITION X,Y:
PRINT #6;CHR$(ABS(F(X,Y))+128*P+16)
22ØØ NEXT I:EX=EX+1:Q1=Q1+1:SOUND Ø,
Ø,Ø,Ø
221Ø POSITION 12,14:PRINT #6;EX;"{2S
PACES}":RETURN
3ØØØ DATA 1,Ø,-1,Ø,Ø,1,Ø,-1
4ØØØ FOR Q1=1ØØ TO 2ØØ STEP 1Ø:FOR I
=Q1+2Ø TO Q1-2Ø STEP -2:SOUND Ø,I,1Ø
,1Ø:NEXT I:NEXT Q1:SOUND Ø,Ø,Ø,Ø:RET
URN

READY !
```

Wenn Sie dann so weit sind, daß Sie ähnlich gute oder vielleicht noch bessere Programme schreiben können, dann hat sich Ihr Computer bereits amortisiert.
Denn: Die Redaktion CHIP-Special kauft sämtliche Programme auf, die unseren Kriterien standhalten.

Wir halten Ihnen die Daumen!

Danger-Snake

Beschreibung des Programmes (Atari 600XL/800XL, 3650 Bytes, zwei Joysticks), für zwei Spieler

Bei diesem Geschicklichkeitsspiel kommt es darauf an, daß sich die Wege der beiden Schlangen nicht kreuzen. Sobald Sie mit Ihrem Endloswurm auf die Schlangenlinie Ihres Gegenspielers treffen, markiert ein Sprite an der Kollisionsstelle den Fehler.

Der Computer zeigt den Gewinner der jeweiligen Runde an und addiert ihm automatisch zehn Punkte auf sein Spielkonto.

Zum Spielverlauf: Sobald das Programm gestartet wurde und die Namen der beiden Spieler eingegeben werden, müssen beide Spieler ihren Joystick bewegen, damit das Spiel starten kann. Vorsicht! Die Schlangen bleiben nicht stehen, wenn Sie den Joystick loslassen. Sie laufen in die Richtung, in die der Joystick zuletzt bewegt wurde.

Hier einige Tips: Diagonale Schlangenlinien können mit etwas Geschick gekreuzt werden. Eine Flucht der Schlange über die Bildschirmbegrenzung hinaus, läßt diese auf der gegenüberliegenden Bildschirmseite wieder erscheinen. Stefan Schwarz

```
100 REM *DANGER-SNAKE*
101 REM *FUER CHIP-SPECIAL*
102 REM *VON STEFAN SCHWARZ*
300 DIM S1$(10),S2$(10),A$(10)
310 GRAPHICS 2:SETCOLOR 2,0,0
320 POSITION 3,5:? #6;"Da{CODE238}GE
r-{CODE243}nA{CODE235}e"
350 ? "NAME DES 1.SPIELERS";:INPUT S
1$
360 ? "NAME DES 2.SPIELERS";:INPUT S
2$
400 DIM S1(30),S2(30)
440 SPIEL=SPIEL+1
450 GRAPHICS 5:SETCOLOR 2,0,0:POKE 7
55,0
455 ? "{11SPACES}SPIEL:{2SPACES}";SP
IEL
456 ? "{11SPACES}";S1$;" :";P1
457 ? "{11SPACES}";S2$;" :";P2
460 S1(3)=19:S1(4)=59:S2(3)=19:S2(4)
=19:COLOR 2:PLOT S1(3),S2(3):COLOR 1
:PLOT S1(4),S2(4)
470 Z=4
480 IF STICK(0)=15 OR STICK(1)=15 TH
EN 480
490 IF Z=24 THEN GOSUB 2300
500 SL=S1(Z-1):SB=S2(Z-1)
510 Z=Z+1:IF Z/2=INT(Z/2) THEN T=0
520 IF Z/2<>INT(Z/2) THEN T=1
530 E=STICK(T):SOUND 0,60,10,8
550 IF E=14 THEN SB=SB-1
560 IF E=13 THEN SB=SB+1
570 IF E=11 THEN SL=SL-1
580 IF E=7 THEN SL=SL+1
590 IF E=10 THEN SL=SL-1:SB=SB-1
600 IF E=5 THEN SL=SL+1:SB=SB+1
610 IF E=9 THEN SL=SL-1:SB=SB+1
620 IF E=6 THEN SL=SL+1:SB=SB-1
630 IF E=15 THEN GOSUB 2000
640 IF SL>77 THEN SL=2
650 IF SB>37 THEN SB=2
660 IF SL<2 THEN SL=77
670 IF SB<2 THEN SB=37
680 LOCATE SL,SB,KONTROLLE
685 IF T=0 AND KONTROLLE=2 THEN 1500
690 IF T=1 AND KONTROLLE=1 THEN 1500
700 COLOR T+1:PLOT SL,SB
705 SOUND 0,121,10,8:POKE 77,0
710 S1(Z)=SL:S2(Z)=SB
720 GOTO 490
1500 GRAPHICS 5+16+32:SETCOLOR 2,0,8
:COLOR 3:GH=0
1505 GH=GH+1
1510 PLOT SL,SB:DRAWTO 2,2
1520 PLOT SL,SB:DRAWTO 77,37
1530 PLOT SL,SB:DRAWTO 2,37
1540 PLOT SL,SB:DRAWTO 77,2
1550 PLOT SL,SB:DRAWTO 38,2
1560 PLOT SL,SB:DRAWTO 38,37
1570 PLOT SL,SB:DRAWTO 2,18
1580 PLOT SL,SB:DRAWTO 77,18
1590 IF GH=2 THEN 1600
1595 COLOR 0:FOR X=1 TO 20:NEXT X:GO
TO 1505
1600 GRAPHICS 18:FOR X=1 TO 255 STEP
 2:SOUND 0,X,6,10:SETCOLOR 4,0,X:NEX
T X
1620 GRAPHICS 2:SETCOLOR 2,0,0
1630 IF T=1 THEN A$=S1$:P1=P1+10
1640 IF T=0 THEN A$=S2$:P2=P2+10
```

```
165Ø POSITION 9-((LEN(A$))/2),3:? #6
;A$
166Ø POSITION 3,5:? #6;"HAT GEWONNEN
!"
167Ø POKE 755,Ø:? :SOUND Ø,Ø,Ø,Ø
168Ø ? " BEIDE ROTEN KNOPF DUECKEN!"
169Ø IF STRIG(Ø)+STRIG(1)=Ø THEN GOT
O 44Ø
17ØØ GOTO 169Ø
2ØØØ U=Ø:J=Ø
2ØØ1 IF S1(Z-2)=2 AND S1(Z-4)=77 THE
N SL=SL+1:U=1
2ØØ2 IF S1(Z-2)=77 AND S1(Z-4)=2 THE
N SL=SL-1:J=1
2ØØ3 IF S2(Z-2)=2 AND S2(Z-4)=37 THE
N SB=SB+1:U=U+2
2ØØ4 IF S2(Z-2)=37 AND S2(Z-4)=2 THE
N SB=SB-1:J=J+2
2ØØ5 IF J<>1 AND J<>3 AND S1(Z-2)>S1
(Z-4) THEN SL=SL+1
2Ø1Ø IF U<>1 AND U<>3 AND S1(Z-2)<S1
(Z-4) THEN SL=SL-1
2Ø2Ø IF J<>2 AND J<>3 AND S2(Z-2)>S2
(Z-4) THEN SB=SB+1
2Ø3Ø IF U<>2 AND U<>3 AND S2(Z-2)<S2
(Z-4) THEN SB=SB-1
2Ø4Ø RETURN
23ØØ S1(Ø)=S1(2Ø):S1(1)=S1(21)
231Ø S1(2)=S1(22):S1(3)=S1(23)
232Ø S2(Ø)=S2(2Ø):S2(1)=S2(21)
233Ø S2(2)=S2(22):S2(3)=S2(23)
234Ø Z=3:RETURN

  READY !
```

Jackpot

Beschreibung des Programmes (Atari 600XL/800 XL/800, Diskette möglich, 9915 Bytes)
"Jackpot" ist die Simulation eines Geldspielautomaten. Nach dem Titel kommt man mit "J" zu den Spielregeln und zum Laden eines alten Spieles, mit "Nein" zum Spiel selbst.
Der Spieler hat die Möglichkeit, ein bis fünf Mark in den Automaten zu werfen, indem er bei der Frage "Wieviel" auf die entsprechende Zahl drückt. Null Mark Einsatz bedeutet Spielende und bietet dem Spieler die Möglichkeit, das Spiel abzuspeichern. Soll das Spiel abgespeichert werden, dann darf die Diskette nicht schreibgeschützt und das Programm auf der Diskette nicht gelockt sein.
Nach jedem Gewinn kann das eingenommene Geld noch einmal riskiert werden. Bei der Wahl von zehn Prozent wird der Gewinn verzehnfacht, bei 30 Prozent wird er verdreifacht und bei 60 Prozent hat man verloren. Pro verlorenes Spiel addiert der Computer 50 Pfennige im Jackpot, der mit drei Jokern gewonnen werden kann. Der Spieler beginnt mit 50 Mark. Das Spiel ist zu Ende, sobald dieser Einsatz verspielt ist.
Gewinnkombinationen:
3xJoker=Jackpot(mal Einsatz)
3xApfel=10 Mark(mal Einsatz)
3xFeige=10 Mark(mal Einsatz)
3xBirne=10 Mark(mal Einsatz)
(FRUCHT)(FRUCHT)(JOKER) gibt zehn Mark, wenn beide Früchte gleich sind.
(BONUS)(BONUS)--- gibt drei Mark.
(BONUS)(-----)--- gibt zwei Mark.

Bernd Moser

```
1ØØ REM * JACKPOT *
1Ø1 REM *FUER CHIP-SPECIAL*
1Ø2 REM *VON BERND MOSER*
18Ø GRAPHICS Ø
19Ø DIM R$(17),G$(8),LE23$(24),LE16$
(17),LE3Ø$(3Ø),X$(5),Y$(5),Z$(5),V$(
8)
2ØØ SETCOLOR 2,Ø,1Ø:SETCOLOR 1,Ø,5
21Ø POKE 752,51
22Ø P=Ø
23Ø R$="RISIKOSPIEL (J/N)"
24Ø G$="GEWINN:"
25Ø K=5Ø
26Ø J=Ø
```

```
270 L=Ø
280 FOR OFT=1 TO 23:LE23$(OFT,OFT)=C
HR$(32):NEXT OFT
290 FOR OFT=1 TO 16:LE16$(OFT,OFT)=C
HR$(32):NEXT OFT
300 FOR OFT=1 TO 3Ø:LE3Ø$(OFT,OFT)=C
HR$(32):NEXT OFT
310 V$="VERLOREN"
320 GOTO 187Ø
330 GOTO 225Ø
390 IF K<=Ø THEN POSITION 19,1:? " P
LEITE!!!":GOSUB 33ØØ
400 IF K<=Ø THEN POSITION 7,11:? " *
* SPIELENDE **"
410 IF K<=Ø THEN FOR WA=1 TO 15ØØ:NE
XT WA:? CHR$(125):END
420 POSITION 2Ø,1:? "{8SPACES}":POSI
TION 2Ø,1:? K;" DM"
430 P=P+1:POSITION 2,17:? "SPIEL: ";
P
440 POSITION 22,19:? "{12SPACES}":PO
SITION 22,19:? J;" DM"
450 POSITION 2,19:? "GELD EINWERFEN.
..":POSITION 2,21:? "WIEVIEL ?"
460 OPEN #1,4,Ø,"K:"
470 GET #1,A
480 CLOSE #1
490 IF A<48 OR A>53 THEN POSITION 2,
19:? LE16$
500 IF A<48 OR A>53 THEN POSITION 2,
19:? "NUR 1-5 DM !!!":GOTO 46Ø
510 IF A=48 THEN 175Ø
520 POSITION 2,19:? CHR$(A);" DM EIN
GESETZT":POSITION 2,21:? LE16$
530 IF A=49 THEN M=1:GOTO 58Ø
540 IF A=5Ø THEN M=2:GOTO 58Ø
550 IF A=51 THEN M=3:GOTO 58Ø
560 IF A=52 THEN M=4:GOTO 58Ø
570 IF A=53 THEN M=5
580 L=M
590 FOR O=1 TO 7
600 POSITION 5,7:? "{5SPACES}":POSIT
ION 17,7:? "{5SPACES}":POSITION 29,7
:? "{5SPACES}"
610 FOR N=1 TO 5Ø:NEXT N
620 POSITION 5,7:? "*****":POSITION
17,7:? "*****":POSITION 29,7:? "****
*"
630 FOR N=1 TO 5Ø:NEXT N
640 NEXT O
650 G=Ø
660 GOSUB 88Ø
670 WALZE1=W
680 GOSUB 88Ø
690 WALZE2=W
700 GOSUB 88Ø
710 WALZE3=W
720 W=WALZE1
730 GOSUB 9ØØ
740 GOSUB 319Ø
750 POSITION 5,7:? X$
760 FOR WA=1 TO 5Ø:NEXT WA
770 W=WALZE2
780 GOSUB 97Ø
790 GOSUB 319Ø
800 POSITION 17,7:? Y$
810 FOR WA=1 TO 5Ø:NEXT WA
820 W=WALZE3
830 GOSUB 1Ø4Ø
840 GOSUB 319Ø
850 POSITION 29,7:? Z$
860 FOR WA=1 TO 5Ø:NEXT WA
870 GOTO 116Ø
880 W=INT(RND(Ø)*5)+1
890 RETURN
900 ON WALZE1 GOTO 91Ø,92Ø,93Ø,94Ø,9
5Ø
910 X$="JOKER":GOTO 96Ø
920 X$="BONUS":GOTO 96Ø
930 X$="BIRNE":GOTO 96Ø
940 X$="FEIGE":GOTO 96Ø
950 X$="APFEL"
960 RETURN
970 ON WALZE2 GOTO 98Ø,99Ø,1ØØØ,1Ø1Ø
,1Ø2Ø
980 Y$="JOKER":GOTO 1Ø3Ø
990 Y$="BONUS":GOTO 1Ø3Ø
1000 Y$="BIRNE":GOTO 1Ø3Ø
1010 Y$="FEIGE":GOTO 1Ø3Ø
1020 Y$="APFEL"
1030 RETURN
1040 ON WALZE3 GOTO 1Ø5Ø,1Ø6Ø,1Ø7Ø,1
Ø8Ø,1Ø9Ø
1050 Z$="JOKER":GOTO 11ØØ
1060 Z$="BONUS":GOTO 11ØØ
1070 Z$="BIRNE":GOTO 11ØØ
1080 Z$="FEIGE":GOTO 11ØØ
1090 Z$="APFEL"
1100 RETURN
1110 GOSUB 319Ø
1120 GOSUB 319Ø
1130 GOSUB 319Ø
1160 REM
1190 IF X$="JOKER" THEN 123Ø
1200 IF X$="BONUS" THEN 127Ø
1210 IF X$=Y$ THEN 129Ø
1220 GOTO 131Ø
1230 IF X$=Y$ THEN 125Ø
1240 GOTO 131Ø
1250 IF Y$=Z$ THEN 133Ø
1260 GOTO 131Ø
1270 IF X$=Y$ THEN 144Ø
1280 GOTO 146Ø
1290 IF Y$=Z$ THEN 148Ø
```

```
1300 IF Z$="JOKER" THEN 1480
1310 POSITION 7,11:? V$:GOSUB 2980:K
=K-M:J=J+Ø.5
1320 FOR WA=1 TO 200:NEXT WA:POSITIO
N 7,11:? LE23$:GOTO 390
1330 FOR OFT=1 TO 5:POSITION 7,11:?
"** JACKPOT **":FOR WA=1 TO 100:NEXT
 WA
1340 POSITION 7,11:? "{3SPACES}JACKP
OT{3SPACES}":FOR WA=1 TO 100:NEXT WA
:NEXT OFT
1350 SOUND Ø,121,1Ø,1Ø
1360 FOR A=Ø TO 25Ø:NEXT A
1370 SOUND 1,96,1Ø,1Ø
1380 FOR A=Ø TO 25Ø:NEXT A
1390 SOUND 2,81,1Ø,1Ø
1400 FOR A=Ø TO 5ØØ:NEXT A
1410 SOUND Ø,Ø,Ø,Ø:SOUND 1,Ø,Ø,Ø:SOU
ND 2,Ø,Ø,Ø
1420 POSITION 7,11:? G$;" JACKPOT *
";M;" DM":GOSUB 3Ø8Ø:M=M*J:J=Ø
1430 GOTO 149Ø
1440 POSITION 7,11:? G$;"3 DM * ";M;
" DM":GOSUB 3Ø8Ø:M=M*3
1450 GOTO 149Ø
1460 POSITION 7,11:? G$;"2 DM * ";M;
" DM":GOSUB 3Ø8Ø:M=M*2
1470 GOTO 149Ø
1480 POSITION 7,11:? G$;"1Ø DM * ";M
;" DM":GOSUB 3Ø8Ø:M=M*1Ø
1490 FOR WA=1 TO 4ØØ:NEXT WA:POSITIO
N 7,11:? LE23$
1500 POSITION 7,11:? LE23$:POSITION
7,11:? G$;" ";M;" DM"
1560 POSITION 7,13:? R$
1570 OPEN #1,4,Ø,"K:"
1580 GET #1,A
1590 CLOSE #1
1600 IF A=74 THEN 163Ø
1610 POSITION 7,11:? LE23$:POSITION
7,13:? LE23$
1620 K=K+M:GOTO 39Ø
1630 POSITION 7,11:? LE23$:POSITION
7,13:? LE23$
1640 RIS=INT(1Ø*RND(Ø)+1)
1650 IF RIS=1 THEN POSITION 7,11:? "
** 1ØFACH **"
1660 IF RIS=1 THEN POSITION 7,13:? G
$;" ";1Ø*M;" DM":K=K+(1Ø*M):GOSUB 3Ø
8Ø
1670 IF RIS=Ø THEN GOTO 173Ø
1680 IF RIS>1 AND RIS<5 THEN POSITIO
N 7,11:? "** 3FACH **"
1690 IF RIS>1 AND RIS<5 THEN POSITIO
N 7,13:? G$;" ";3*M;" DM":K=K+(3*M)
1700 IF RIS>1 AND RIS<5 THEN GOSUB 3
Ø8Ø:GOTO 173Ø
1710 IF RIS>4 AND RIS<11 THEN POSITI
ON 7,11:? "** VERLOREN **"
1720 IF RIS>4 AND RIS<11 THEN GOSUB
298Ø:K=K-L:J=J+Ø.5
1730 FOR WA=1 TO 4ØØ:NEXT WA:POSITIO
N 7,11:? LE23$
1740 POSITION 7,13:? LE23$:GOTO 39Ø
1750 ? CHR$(125):SETCOLOR 2,4,4:SETC
OLOR 1,Ø,1Ø:? :? :?
1760 ? "{3SPACES}SPIELENDE "
1770 ? :? "Ihr Kapital betraegt ";K;
" DM."
1780 ? "Im Jackpot sind ";J;" DM."
1790 ? :? :? "Wollen Sie diesen Spie
lstand ab-":? "speichern? (J/N)"
1800 OPEN #1,4,Ø,"K:"
1810 GET #1,A
1820 CLOSE #1
1830 IF A=74 THEN 285Ø
1840 END
1870 REM
1910 POSITION 2,1:? "1-5 DM{5SPACES}
";CHR$(124)
1920 POSITION 2,2:? "EINWERFEN{2SPAC
ES}";CHR$(124)
1930 POSITION 18,Ø:? CHR$(17);:FOR O
FT=1 TO 14:? CHR$(18);:NEXT OFT:? CH
R$(5)
1940 POSITION 18,1:? CHR$(1):POSITIO
N 33,1:? CHR$(4)
1950 POSITION 18,2:? CHR$(26);:FOR O
FT=1 TO 14:? CHR$(18);:NEXT OFT:? CH
R$(3)
1960 POSITION 2,5:FOR OFT=1 TO 35:?
CHR$(16Ø);:NEXT OFT
1970 POSITION 13,5:? CHR$(32):POSITI
ON 25,5:? CHR$(32)
1980 POSITION 2,9:FOR OFT=1 TO 35:?
CHR$(16Ø);:NEXT OFT
1990 POSITION 13,9:? CHR$(32):POSITI
ON 25,9:? CHR$(32)
2000 POSITION 2,6:? CHR$(14);:FOR OF
T=1 TO 9:? CHR$(13);:NEXT OFT
2010 ? CHR$(14);CHR$(32);CHR$(14);
2020 FOR OFT=1 TO 9:? CHR$(13);:NEXT
 OFT
2030 ? CHR$(14);CHR$(32);CHR$(14);:F
OR OFT=1 TO 9:? CHR$(13);:NEXT OFT:?
 CHR$(14)
2040 POSITION 2,8:? CHR$(13);:FOR OF
T=1 TO 9:? CHR$(14);:NEXT OFT
2050 ? CHR$(13);CHR$(32);CHR$(13);
2060 FOR OFT=1 TO 9:? CHR$(14);:NEXT
 OFT:? CHR$(13);CHR$(32);CHR$(13);
2070 FOR OFT=1 TO 9:? CHR$(14);:NEXT
 OFT:? CHR$(13)
2080 POSITION 2,7:? CHR$(2):POSITION
```

```
 12,7:? CHR$(22)
2Ø9Ø POSITION 14,7:? CHR$(2):POSITIO
N 24,7:? CHR$(22)
21ØØ POSITION 26,7:? CHR$(2):POSITIO
N 36,7:? CHR$(22)
211Ø POSITION 5,7:? "*****"
212Ø POSITION 17,7:? "*****"
213Ø POSITION 29,7:? "*****"
214Ø POSITION 2Ø,15:? "*************
**"
215Ø POSITION 2Ø,16:? "****JACKPOT**
**"
216Ø POSITION 2Ø,17:? "*************
**"
217Ø POSITION 2Ø,18:? "*":POSITION 3
4,18:? "*"
218Ø POSITION 2Ø,19:? "*":POSITION 3
4,19:? "*"
219Ø POSITION 2Ø,2Ø:? "* -----------
 *"
22ØØ POSITION 2Ø,21:? "*":POSITION 3
4,21:? "*"
221Ø POSITION 2Ø,22:? "*************
**"
222Ø GOTO 33Ø
225Ø REM
228Ø FOR OFT=1 TO 3:POSITION 7,11:?
"JACKPOT GELDSPIELAUTOMAT"
229Ø FOR WA=1 TO 1ØØ:NEXT WA:POSITIO
N 7,11
23ØØ ? "(C) 1983 BY BERND MOSER ":FO
R WA=1 TO 1ØØ:NEXT WA:NEXT OFT
231Ø POSITION 7,11:? LE23$:POSITION
7,11:? "KENNEN SIE DIE SPIELREGELN"
232Ø POSITION 7,13:? LE23$:POSITION
7,13:? "UND GEWINNKOMBINATIONEN (J/N
)";
233Ø OPEN #1,4,Ø,"K:"
234Ø GET #1,A
235Ø CLOSE #1
236Ø IF A<>74 THEN 238Ø
237Ø POSITION 7,11:? LE3Ø$:POSITION
7,13:? LE3Ø$:GOTO 39Ø
238Ø GRAPHICS Ø:POKE 752,1:SETCOLOR
2,2,Ø
239Ø ? :? "{1ØSPACES}SPIELREGELN ZUM
":? :? "{4SPACES}*JACKPOT GELDSPIELA
UTOMAT*"
24ØØ ? :? :? "Sie koennen 1-5 DM ein
werfen, indem"
241Ø ? "Sie bei der Frage 'Wieviel'
eine Zahl":? "druecken."
242Ø ? :? "Nach jedem Gewinn koennen
 Sie das "
243Ø ? "Risikospiel machen. Druecken
 Sie J/N"
244Ø ? "bei der Frage 'Risiko?'. Sie
 erhalten dann den 1Øfachen (1Ø%), 3
fachen"
245Ø ? "(3Ø%) oder Øfachen (6Ø%) Gew
inn."
246Ø ? :? "Pro verlorenes Spiel addi
ert der":? "Computer -.5Ø Pf im Jack
pot."
247Ø ? :? "Ø DM Einsatz bedeutet Spi
elende."
248Ø ? :? :? "Bitte eine Taste druec
ken..."
249Ø IF PEEK(764)=255 THEN 249Ø
25ØØ POKE 764,255
251Ø ? CHR$(125):? :? "{2SPACES}* GE
WINNKOMBINATIONEN *"
252Ø ? :? "{2SPACES}3 X JOKER{2SPACE
S}={2SPACES}JACKPOT(mal Einsatz)"
253Ø ? :? "{2SPACES}3 X APFEL{2SPACE
S}={2SPACES}1Ø.-DM (mal Einsatz)"
254Ø ? :? "{2SPACES}3 X FEIGE{2SPACE
S}={2SPACES}1Ø.-DM (mal Einsatz)"
255Ø ? :? "{2SPACES}3 X BIRNE{2SPACE
S}={2SPACES}1Ø.-DM (mal Einsatz)":?
256Ø ? "<FRUCHT><FRUCHT><JOKER> gibt
 1Ø.-DM"
257Ø ? "wenn beide Fruechte gleich s
ind."
258Ø ? :? "<BONUS><BONUS><-----> gib
t 3.- DM."
259Ø ? :? "<BONUS><-----><-----> gib
t 2.- DM."
26ØØ ? :? :? "Bitte eine Taste druec
ken..."
261Ø IF PEEK(764)=255 THEN 261Ø
262Ø POKE 764,255
263Ø ? CHR$(125):? :?
264Ø ? "Dieses Programm sollte auf d
er"
265Ø ? "Diskette nicht gelockt sein,
 da es"
266Ø ? "sonst nicht abzuspeichern is
t."
267Ø ? :? :? "Ist dies ein altes Spi
el (J/N)"
268Ø OPEN #1,4,Ø,"K:"
269Ø GET #1,A
27ØØ CLOSE #1
271Ø ? CHR$(125)
272Ø IF A=74 THEN 279Ø
273Ø SETCOLOR 2,Ø,1Ø:SETCOLOR 1,Ø,5:
GOTO 187Ø
279Ø RESTORE
28ØØ READ K,J,P
281Ø DATA 5Ø
282Ø DATA Ø
283Ø DATA 1
284Ø P=P-1:SETCOLOR 2,Ø,1Ø:SETCOLOR
```

```
1,Ø,5:GOTO 187Ø
285Ø ? CHR$(125):POKE 752,Ø
286Ø POSITION 2,7:? "981Ø DATA ";K
287Ø POSITION 2,8:? "982Ø DATA ";J
288Ø POSITION 2,9:? "983Ø DATA ";P
289Ø POSITION Ø,Ø:POKE 842,13
29ØØ POSITION 2,12:? "CONT"
291Ø POSITION 2,5:STOP
292Ø POKE 842,12
293Ø ? "PROGRAMM WIRD ABGESAVED"
294Ø SAVE "D:JACKPOT"
295Ø ? :? "ENDE":END
298Ø REM
3Ø1Ø FOR Z=3Ø TO 2ØØ
3Ø2Ø SOUND 1,Z,1Ø,4
3Ø3Ø NEXT Z
3Ø4Ø SOUND 1,Ø,Ø,Ø
3Ø5Ø RETURN
3Ø8Ø REM
311Ø FOR A=1 TO 1Ø
312Ø FOR I=1 TO 2Ø
313Ø SOUND 2,I,1Ø,4
314Ø NEXT I:NEXT A
315Ø SOUND 2,Ø,Ø,Ø
316Ø RETURN
319Ø REM
322Ø FOR A=Ø TO 1Ø
323Ø SOUND Ø,8Ø,A,4
324Ø FOR X=Ø TO 2
325Ø NEXT X:NEXT A
326Ø SOUND Ø,Ø,Ø,Ø
327Ø RETURN
333Ø FOR A=Ø TO 15Ø
334Ø SOUND Ø,A,1Ø,1Ø
335Ø FOR Z=Ø TO 5:NEXT Z
336Ø NEXT A
337Ø SOUND Ø,Ø,Ø,Ø
338Ø RETURN

 READY !
```

Kanopus

Beschreibung des Programmes (Atari 600 XL/800 XL, 8360 Bytes, zwei Joysticks, für zwei Spieler)
Ziel dieses Spiels ist es, auf dem schnellsten Weg vom Planeten Saturn zum Stern Kanopus zu fliegen. Um diesen Stern möglichst rasch zu erreichen, müssen Sie - soweit möglich - der Kompaßnadel nachfliegen. Der Computer schreibt nämlich den Weg vor. Von jedem Stern führen stets vier Flugbahnen weg, die nie alle die gleiche Farbe haben.
Drücken Sie nun auf den Knopf am Joystick. Es folgt die Weganweisung. Wählen Sie jetzt den Weg, der Ihnen am günstigsten erscheint. Vorsicht vor schwarzen Löchern: Wenn Sie da hineingeraten, erleben Sie Ihr blaues Wunder. Sie können um Lichtjahre zurückgeworfen werden. Aber Sie können auch vorgeschleudert werden, wenn Sie Glück haben, bis ins Ziel. Falls Sie jedoch nicht in das schwarze Loch hineinfliegen wollen, so fallen Sie automatisch um zwei Lichtjahre zurück. Dafür dürfen Sie noch einen Zug machen. Wer zuerst Kanopus erreicht, ist Sieger!

Stefan Schwarz

```
1 REM * KANOPUS *
2 REM * FUER CHIP-SPECIAL*
3 REM * VON STEFAN SCHWARZ*
2Ø GRAPHICS 2+16:DIM JA$(5)
3Ø POSITION 2,2:? #6;"NUN KOMMT DAS
..."
4Ø POSITION 2,4:? #6;" KANOPUS SPIEL
"
42 POSITION 2,7:? #6;"start {CODE228
}Ru{CODE229}Ck{CODE229}n!"
45 FOR X=1 TO 4ØØ:NEXT X
5Ø IF PEEK(53279)=6 THEN 9Ø
55 READ Q,F
6Ø IF Q=-1 OR F=-1 THEN RESTORE 5ØØØ
:GOTO 5Ø
7Ø SOUND Ø,Q,1Ø,8
75 FOR RTY=Ø TO F:NEXT RTY
8Ø GOTO 5Ø
9Ø SOUND Ø,Ø,Ø,Ø
1ØØ REM *** KANOPUS-SPIEL ***
```

```
110 DIM S1$(10),S2$(10)
120 GRAPHICS 2
130 POSITION 5,1:? #6;"KANOPUS"
140 POSITION 5,3:? #6;"{3SPACES}-{3S
PACES}"
150 POSITION 5,5:? #6;" SPIEL "
160 SETCOLOR 2,0,0
170 ? "NAME DES 1.SPIELERS";:INPUT S
1$
180 ? "NAME DES 2.SPIELERS";:INPUT S
2$
200 ZUG=4
210 DIM WIN$(10),KAN$(1)
300 SETCOLOR 0,3,10
310 SETCOLOR 1,8,10
320 DIM X(4),LJ0(400),T0(400)
340 DIM KO$(2),W(4),W$(4)
360 OPEN #1,4,0,"K:"
400 LJ0(3)=15:T0(3)=1:LJ0(4)=15:T0(4
)=1
405 GOSUB 6000
410 GOTO 3000
420 IF KO=1 THEN KO$="{CODE27/28}"
440 IF KO=3 THEN KO$="{CODE27/30}"
450 IF KO=4 THEN KO$="{CODE27/31}"
460 RETURN
700 IF ZG=0 THEN 2600
705 FOR C=1 TO 4
710 CO=INT(RND(0)*2)+2
720 X(C)=CO
730 NEXT C
740 IF X(1)=X(2) AND X(2)=X(3) AND X
(3)=X(4) THEN 700
800 GRAPHICS 5
820 A=RND(0)*75+2
830 B=RND(0)*35+2
840 PLOT A,B
850 COLOR X(1):DRAWTO 0,20
860 PLOT A,B
870 COLOR X(2):DRAWTO 79,20
880 PLOT A,B
890 COLOR X(3):DRAWTO 39,0
900 COLOR 1:PLOT A,B
910 COLOR X(4):DRAWTO 39,39
920 RETURN
930 ZUG=ZUG+1:? CHR$(125)
940 IF ZUG/2=INT(ZUG/2) THEN TE=1
950 IF ZUG/2<>INT(ZUG/2) THEN TE=0
960 IF TE=0 THEN ZUG0=ZUG0+1
970 IF TE=1 THEN ZUG1=ZUG1+1
980 IF TE=0 THEN 1000
990 IF TE=1 THEN 1050
1000 ? " ";S1$;"'S ";ZUG0;".ZUG"
1010 ? " SIE SIND ";LJ;" LICHTJAHRE
ENTFERNT"
1020 ? " KOMPASSANZEIGE: ";KO$
1030 GOTO 1100
1050 ? " ";S2$;"'S{2SPACES}";ZUG1;".
ZUG"
1060 ? " SIE SIND ";LJ;" LICHTJAHRE
ENTFERNT"
1070 ? " KOMPASSANZEIGE: ";KO$
1100 RETURN
1110 REM *ZUFALLS GENERATOR*
1140 WEE=INT(RND(0)*2)+1
1160 IF WEE=1 THEN W$="GELB"
1170 IF WEE=2 THEN W$="BLAU"
1200 RETURN
1210 REM *STEUERN*
1220 IF STICK(TE)=15 THEN 1220
1230 IF STICK(TE)=14 AND STRIG(TE)=0
 THEN 1270
1240 IF STICK(TE)=13 AND STRIG(TE)=0
 THEN 1330
1250 IF STICK(TE)=11 AND STRIG(TE)=0
 THEN 1390
1260 IF STICK(TE)=7 AND STRIG(TE)=0
THEN 1450
1265 IF H<>14 AND H<>13 AND H<>7 AND
 H<>11 THEN 1220
1270 LOCATE 39,0,FLU:IF FLU=WEE+1 TH
EN 1300
1280 GOSUB 2500
1290 FOR X=1 TO 900:NEXT X:GOTO 3150
1300 IF KO=1 THEN LJ=LJ-1
1305 IF KO=3 THEN LJ=LJ+0.5
1310 IF KO=4 THEN LJ=LJ+0.5
1315 COLOR 1:PLOT A,B:DRAWTO 39,0
1320 RETURN
1330 LOCATE 39,39,FLU:IF FLU=WEE+1 T
HEN 1360
1340 GOSUB 2500
1350 FOR X=1 TO 900:NEXT X:GOTO 3150
1360 IF KO=1 THEN LJ=LJ+1
1365 IF KO=3 THEN LJ=LJ+1.5
1370 IF KO=4 THEN LJ=LJ+1.5
1375 COLOR 1:PLOT A,B:DRAWTO 39,39
1380 RETURN
1390 LOCATE 0,20,FLU:IF FLU=WEE+1 TH
EN 1420
1400 GOSUB 2500
1410 FOR X=1 TO 900:NEXT X:GOTO 3150
1420 IF KO=1 THEN LJ=LJ+0.5:KO=4:GOT
O 1445
1425 IF KO=3 THEN LJ=LJ-0.5:KO=1:GOT
O 1445
1430 IF KO=4 THEN LJ=LJ+0.5
1445 COLOR 1:PLOT A,B:DRAWTO 0,20:RE
TURN
1450 LOCATE 79,20,FLU:IF FLU=WEE+1 T
HEN 1480
1460 GOSUB 2500
1470 FOR X=1 TO 900:NEXT X:GOTO 3150
1480 IF KO=1 THEN LJ=LJ+0.5:KO=3:GOT
```

```
O 1520
1490 IF KO=4 THEN LJ=LJ-Ø.5:KO=1:GOT
O 1520
1500 IF KO=3 THEN LJ=LJ+Ø.5
1520 COLOR 1:PLOT A,B:DRAWTO 79,20:R
ETURN
2000 FOR X=255 TO Ø STEP -1
2010 SOUND Ø,X,10,8
2020 NEXT X
2030 SOUND Ø,121,10,8
2040 FOR X=1 TO 70:NEXT X
2050 SOUND Ø,Ø,Ø,Ø
2060 RETURN
2500 ? CHR$(125):? " DIESEN WEG KOEN
NEN SIE NICHT NEHMEN"
2510 FOR X=1 TO 20
2520 SOUND Ø,121,10,8
2530 FOR SY=1 TO 15:NEXT SY
2540 SOUND Ø,60,10,8
2550 FOR SY=1 TO 15:NEXT SY
2560 NEXT X
2570 SOUND Ø,Ø,Ø,Ø
2580 RETURN
2600 GRAPHICS 7:SETCOLOR Ø,Ø,8:COLOR
 1
2610 PI=3.14159:SO=Ø
2620 FOR X=Ø TO 2*PI STEP PI/54
2625 SO=SO+1
2627 SOUND Ø,SO,10,8
2630 PLOT COS(X)*20+79,SIN(X)*20+39
2640 NEXT X
2650 SOUND Ø,Ø,Ø,Ø
2660 GOTO 920
2700 ? CHR$(125)
2710 ? "SIE SIND BEI EINEM SCHWARZEN
 LOCH"
2720 ? "WOLLEN SIE IN DIESES HINEIN
FAHREN?"
2730 GET #1,LO:IF LO=ASC("N") THEN L
J=LJ+2:ZG=ZG+1:GOTO 3060
2740 AS=INT(RND(Ø)*2)
2750 IF AS=Ø THEN 2760
2755 IF AS=1 THEN 2820
2760 LJ=INT(LJ/2)
2780 KOZ=INT(RND(Ø)*10)
2790 IF KOZ<5 THEN KO=1
2800 IF KOZ<8 AND KOZ>4 THEN KO=3
2810 IF KOZ>7 THEN KO=4
2813 GOSUB 5100
2815 GOTO 3200
2820 YP=INT(RND(Ø)*6)+1
2830 YH=INT(RND(Ø)*3)+1
2840 IF YH=1 OR YH=2 THEN LJ=LJ-YP
2850 IF YH=3 THEN LJ=LJ+YP
2860 GOTO 2780
3000 ZG=INT(RND(Ø)*6)
3005 GOSUB 700
3010 KO=TØ(ZUG-1):LJ=LJØ(ZUG-1)
3050 GOSUB 420:GOSUB 930
3055 GOTO 3070
3060 GOSUB 700:GOSUB 420:GOSUB 960
3070 IF STRIG(TE)=Ø THEN 3100
3080 GOTO 3070
3100 IF ZG=Ø THEN 2700
3110 GOSUB 1110
3140 ? CHR$(125)
3150 IF WEE=1 THEN ? " NEHMEN SIE EI
NEN GELBEN WEG!"
3160 IF WEE=2 THEN ? " NEHMEN SIE EI
NEN BLAUEN WEG!"
3170 ? "{5SPACES}JOYSTICK BEWEGEN !"
3180 ? " GLEICHZEITIG KNOPF DRUECKEN
!"
3190 GOSUB 1210:GOSUB 420:GOSUB 2000
3200 TØ(ZUG)=KO:LJØ(ZUG)=LJ
3220 IF LJ<Ø.5 THEN 4000
3230 GOTO 3000
4000 GOSUB 6500
4003 GRAPHICS 2
4005 SETCOLOR 2,Ø,Ø
4010 IF TE=Ø THEN WIN$=S1$
4020 IF TE=1 THEN WIN$=S2$
4030 POSITION 9-LEN(WIN$)/2,2:? #6;W
IN$
4040 POSITION 2,4:? #6;" HAT GEWONNE
N !"
4050 FOR X=Ø TO 255 STEP 1.5
4060 SOUND Ø,X,6,8
4062 LI=LI+1
4064 IF LI/2=INT(LI/2) THEN LIH=14
4065 IF LI/2<>INT(LI/2) THEN LIH=Ø
4067 SETCOLOR 4,Ø,LIH
4068 SETCOLOR 2,Ø,LIH
4070 NEXT X
4080 SOUND Ø,Ø,Ø,Ø
4090 ? " WOLLEN SIE NOCH EINMAL SPIE
LEN";:INPUT JA$
4100 IF JA$(1,1)="J" THEN RUN
4120 END
4200 GOTO 4200
5000 DATA 121,150,96,70,81,70,96,70,
91,70,108,70,121,70,Ø,70,121,150,128
,70,144,70,128,70,121,70
5010 DATA 144,70,128,300,-1,-1
5030 DATA 121,144,120,128,162,120,14
4,182,120,162,193,120,182,217,120,16
2,193,120,243,243,120,-1,-1,-1
5100 RESTORE 5030
5110 READ VN,VM,NV
5115 IF VN=-1 OR VM=-1 THEN 5200
5120 SOUND Ø,VN,10,8
5125 SOUND 1,VM,10,8
5130 FOR X=Ø TO NV:NEXT X
5140 GOTO 5110
```

```
5200 SOUND 0,0,0,0:SOUND 1,0,0,0:RET
URN
5570 GRAPHICS X+32
5580 FOR D=1 TO 400:NEXT D
5590 NEXT X
6000 GRAPHICS 7:SETCOLOR 0,3,8
6010 COLOR 1
6020 P=3.14159
6030 FOR X=0 TO 2*P STEP P/54
6050 PLOT COS(X)*22+79,SIN(X)*22+39
6055 DRAWTO 79,39
6060 NEXT X
6080 FOR X=0 TO P STEP P/54
6085 COLOR 3
6090 PLOT COS(X)*23+79,SIN(X)*6+39
6095 COLOR 2
6097 PLOT COS(X)*23+79,SIN(X)*7+39
6100 NEXT X
6110 ? " SIE SIND BEIDE AUF DEM SATU
RN UND"
6120 ? " VERSUCHEN DEN PLANETEN KANO
PUS AUF"
6130 ? " DEM SCHNELLSTEN WEG ZU ERRE
ICHEN"
6140 ? " GUTE REISE ! {CODE210/197/2
12/213/210/206/160/196/210/213/197/1
95/203/197/206}";:INPUT KAN$
6150 RETURN
6500 GRAPHICS 7
6505 P=3.14159
6510 GOTO 6610
6520 COLOR 2
6530 FOR X=0 TO 2*P STEP P/54
6540 PLOT COS(X)*7+39,SIN(X)*7+19
6545 DRAWTO 39,19
6546 NEXT X
6547 FOR D=1 TO 255 STEP 5
6548 SOUND 0,D,10,8
6549 NEXT D
6550 SOUND 0,0,0,0
6555 FOR Y=1 TO 20
6560 FOR X=1 TO 15
6570 SETCOLOR 1,X,6
6590 NEXT X
6600 NEXT Y
6605 RETURN
6610 ? CHR$(125)
6620 ? "{6SPACES}K A N O P U S{4SPAC
ES}I N "
6630 PRINT
6640 ? "{9SPACES}S I C H T ! !"
6650 GOTO 6520

READY !
```

Siebzehn und vier

B II 0-65

Beschreibung des Programmes (Atari 600 XL/800 XL, 4580 Bytes, ein Spieler)
Bei 17 und vier handelt es sich um das auf Computer übertragene Kartenspiel. Sie müssen versuchen, 21 Punkte zu erreichen. Das Ziehen der Karten funktioniert durch Betätigen der START-Taste. Nähere Details werden zu Beginn des Programmes rechtzeitig geklärt. Der Computer hält die Karten der Bank in den Händen und zeigt sie am Ende des Spieles und sagt Ihnen auch, ob und wieviel Sie gewonnen haben. Ihr Startkapital beträgt 1000 Bucks. Die Bank besitzt 250 000 Bucks. Viel Glück!

Wolfgang Schmidt

```
10 REM * 17 + 4 *
11 REM * FUER CHIP-SPECIAL*
12 REM * VON WOLFGANG SCHMIDT*
50 GRAPHICS 0:POKE 710,176:POKE 712,
176:POKE 709,222:POKE 752,1
100 ? "{CODE125}{7SPACES}{CODE96/16/
123/0}{2SPACES}17+4{2SPACES}{CODE0/1
23/16/96}"
110 DIM R$(2),K(8),F(4),W(8),S(8,4),
CB(10)
120 KS=1000:KB=250000:A=-8:B=2:CN=53
279
130 FOR I=1 TO 8:READ Q1,Q2:K(I)=Q1:
W(I)=Q2:FOR J=1 TO 4:S(I,J)=0:NEXT J
:NEXT I
140 FOR I=1 TO 4:READ Q1:F(I)=Q1:NEX
T I
180 DATA 55,7,56,8,57,9,49,10,66,2,6
8,3,75,4,65,11
190 DATA 96,123,0,16
195 DATA 1,1,5,1,1,7,5,7
200 ? :PRINT "Du besitzt 1000 Bucks
```

```
und spielst{5SPACES}gegen die Bank,d
ie 250000 Bucks haelt."
210 ? :PRINT "{CODE210/197/199/197/2
04/206/186}Du erhaelst beliebig viel
e Kar-ten deren gemeinsamer Bonus {C
ODE178/177} nicht{2SPACES}ueberschre
iten darf."
220 ? "Du und der Computer bestimmen
 abwech- selnd ob eine neue Karte ge
wuenscht{3SPACES}wird."
230 ? :? "Der Computer kennt deine K
arten, aber du nicht seine - dafuer
erhaelst du{3SPACES}immer den doppel
ten Gewinn !"
260 ? :? :? "DRUECKE {CODE211/212/19
3/210/212}"
270 IF PEEK(CN)=7 THEN 270
290 ? CHR$(125)
300 REM GAME*
305 POKE 703,10:? "{CODE203/225/240/
233/244/225/236/186} ";KS
310 ? "Wieviele Bucks setzt du";:INP
UT Z:IF Z<=0 THEN 310
320 IF Z>KS THEN ? "KREDITKARTEN WER
DEN NICHT ANGENOMMEN !":GOTO 310
325 GOSUB 700
330 ? "{CODE228/229/233/238/229/160/
203/225/242/244/229/186}"
340 POKE 703,24:B=B+2*(A>31):A=A+8-3
9*(A>31)
350 N=INT(RND(0)*8)+1
360 M=INT(RND(6)*4)+1
370 IF S(N,M)=1 THEN 350
380 S(N,M)=1:GOSUB 1050
390 IF BS+W(N)>21 AND N=8 THEN BS=BS
-10
400 BS=BS+W(N):POSITION 30,1:? "{COD
E194/207/206/213/211/186} ";BS:POSIT
ION 2,12:? ""
410 IF BS>21 THEN 540
450 POKE 703,6:IF BC>BS AND BC>15 TH
EN ? "ICH ZIEHE NICHT MEHR.":GOTO 50
0
460 CC=CC+1:? "ICH ZIEHE NUN MEINE "
;CC;". KARTE..."
470 C=INT(RND(0)*8)+1:G=INT(RND(8)*4
)+1:IF S(C,G)=1 THEN 470
480 S(C,G)=1:BC=BC+W(C)-10*(C=8 AND
BC>9):CB(CC)=C+G/100
490 IF BC>21 THEN ? "DU HAST GEWONNE
N...":KS=KS+2*Z:KB=KB-2*Z:GOSUB 770:
GOTO 600
500 IF F=1 THEN 530
505 ? "Willst du noch einmal ziehen
?{8SPACES}(START=JA/ SELECT=NEIN)"
510 IF PEEK(CN)=7 THEN 510
520 IF PEEK(CN)=6 THEN 330
530 F=1:IF BS>=BC THEN 450
540 ? "{CODE224/160/196/213/160/200/
193/211/212/160/214/197/210/204/207/
210/197/206/160/161}{2SPACES}(BB=";B
C;")":GOSUB 750
550 KS=KS-Z:KB=KB+Z
600 POSITION 2,20:? "{CODE156/156/15
6}"
603 ? "KAPITAL: ";KS,"BANK: ";KB:POK
E 703,24
605 IF KS<=0 THEN ? "{CODE125}BEEHRE
N SIE UNS BALD WIEDER ! ":POKE 752,0
:END
610 IF KB<=0 THEN ? "{CODE96/96/96}
DU HAST DIE {CODE194/160/193/160/206
/160/203} GESPRENGT !!!":END
613 POSITION 0,11:? "DIE KARTEN DER
BANK:"
615 FOR D=1 TO CC:N=INT(CB(D)):M=(CB
(D)-N)*100
620 A=-8+D*8-39*(D>5):B=12+2*(D>5):G
OSUB 1050:NEXT D
640 ? "NEUES SPIEL= {CODE207/208/212
/201/207/206} "
650 FOR I=1 TO 8:FOR J=1 TO 4:S(I,J)
=0:NEXT J:NEXT I
660 IF PEEK(CN)=7 THEN 660
670 IF PEEK(CN)<>3 THEN END
680 CC=0:B=2:F=0:BC=0:BS=0:A=-8:GOTO
 290
700 H=40:FOR L=8 TO 0 STEP -0.2:FOR
V=1 TO C:SOUND 0,H,10,V:NEXT V
710 FOR V=2*L TO 1 STEP -1:SOUND 0,F
,10,V:NEXT V:NEXT L:RETURN
750 FOR H=15 TO 0 STEP -1:FOR R=30 T
O 0 STEP -5:SOUND 0,R,10,H:NEXT R:NE
XT H:RETURN
770 FOR I=230 TO 50 STEP -4:SOUND 0,
I,10,10:NEXT I
780 FOR BI=14 TO 0 STEP -0.1:SOUND 0
,80,10,BI:SOUND 1,81,10,BI:NEXT BI
790 RETURN
900 END
1050 R$=CHR$(K(N)):IF R$="1" THEN R$
="10":LB=1
1100 POSITION A,B:? "{CODE17/18/18/1
8/18/18/5/30}";
1110 FOR I=1 TO 7:POSITION A,B+I:? "
{CODE124}{5SPACES}{CODE124/30}";:SOU
ND 0,I*2,8,7-I
1120 NEXT I:POSITION A,B+8:? "{CODE2
6/18/18/18/18/18/3/30}";
1130 RESTORE 195:FOR I=1 TO 4:READ X
,Y:POSITION A+X-LB*(X=5),B+Y:? R$:NE
XT I
1150 POSITION A+3,B+4:? CHR$(F(M))
1200 LB=0:RETURN
READY !
```

Castle Combat

Beschreibung des Programmes (Atari 600XL/800XL, 2 Joysticks, 7320 Bytes)
Castle Combat wird auf einem 6x6 Felder großen, auf der Mattscheibe abgebildetem Brett gespielt. In 21 Zügen setzen die beiden Spieler abwechselnd jeweils eine Burg auf ein freies Spielfeld, auf eine eigene Fahne oder erweitern eine Burg der eigenen Farbe um eine Stufe. Dabei kann jede Burg allerdings nur bis zur dritten Stufe ausgebaut werden. Darüber hinaus ist keine Erweiterung möglich. Nach jedem Zug wertet der Computer für jedes Feld aus, welcher Spieler dort ein Kräfteübergewicht hat. Dazu werden die Burgstärken in dem betreffenden Feld und den vier waagrecht und senkrecht angrenzenden Feldern in Betracht gezogen.
Hierfür ein Beispiel: In der Abbildung hat Weiß die mit einem Pfeil versehene Burg soeben errichtet. Für das Kräfteverhältnis auf dem mittleren Feld bedeutet das, daß jetzt dort vier schwarze Stärkepunkte wirksam sind, nämlich zwei Punkte für die Burg, die auf diesem Feld steht und weitere zwei Punkte für die Burg links daneben. Der weiße Spieler bringt aber fünf Stärkepunkte, nämlich je zwei für die Burgen rechts und unten und einen für die neu errichtete Burg oben.
Die mittlere, schwarze Burg der Stufe zwei wird also entfernt und eine Fahne des weißen Spielers deklariert die nun geänderten Besitzverhältnisse. Fahnen werden übrigens bei der Kräfteauswertung nicht mitgezählt, sie ändern nichts am Kräfteverhältnis und sind nur dazu da, den Gegner an der Besetzung des jeweiligen Feldes zu hindern.

Wolfgang Schmidt

```
100 REM * CASTLE COMBAT *
101 REM * FUER CHIP-SPECIAL*
102 REM * VON WOLFGANG SCHMIDT*
110 S=1:V=656:H=V+1
120 DIM N$(30),P$(15),F(7,7),X(1),Y(
1),W(2)
130 X(0)=1:Y(0)=1:X(1)=6:Y(1)=6:W(0)
=0:W(2)=0
140 OPEN #1,4,0,"K"
200 REM *SPIELREGELN*
210 PRINT "{CODE125}*** CASTLE COMBA
T ***"
220 PRINT "{CODE29}Name des 1.Spiele
rs: ";:INPUT N$
225 L1=LEN(N$):PRINT "{CODE29}Name d
es 2.Spielers: ";:INPUT P$
230 L2=LEN(P$):N$(L1+1)=P$
240 POSITION 2,0:PRINT "{CODE29/29/1
56/156/156/156}<SPIELREGELN>"
245 PRINT "{CODE29}In 21 Zuegen setz
en die beiden Spielerabwechselnd jew
eils eine Burg auf ein freies Feld o
der erweitern"
250 PRINT "eine schon vorhandene eig
ene Burg um{2SPACES}eine Stufe. Hier
bei kann eine Burg{4SPACES}allerding
s nur bis zur"
255 PRINT "dritten Stufe ausgebaut w
erden. Nach{2SPACES}jedem Zug wertet
 der Computer fuer{4SPACES}jedes Fel
d aus, welcher"
260 PRINT "Spieler dort ein Kraefteu
ebergewicht{2SPACES}hat. Dazu werden
 die Burgstaerken in{2SPACES}dem bet
reffenden Feld"
265 PRINT "in den vier direkt angren
zenden{7SPACES}Feldern in Betracht g
ezogen."
270 PRINT "{CODE29}ALLES KLAR ? - DA
NN BEIDE DEN ROTEN{3SPACES}KNOPF DRU
ECKEN."
290 IF STRIG(0)+STRIG(1) THEN 290
300 REM *FELD-INIT/ SPIELFELD AUFBAU
*
305 PRINT "{CODE125}MOMENT BITTE."
310 FOR I=0 TO 7:FOR K=0 TO 7
320 F(I,K)=9*(I=0 OR K=0 OR I=7 OR K
=7):NEXT K:NEXT I
350 GRAPHICS 7:COLOR 3:POKE 752,1:PO
KE 708,4
360 FOR I=0 TO 78 STEP 13:PLOT 0,I:D
RAWTO 156,I:NEXT I
370 FOR I=0 TO 157 STEP 26:PLOT I,0:
DRAWTO I,77:NEXT I
```

```
380 REM F(1,1)=-1
400 REM *HAUPTSPIEL*
410 Z=Z+1:IF Z>=22 THEN 1000
420 S=S+1:IF S>1 THEN S=0
430 POKE V,0:POKE H,2:PRINT "{CODE12
5}";N$(L1*S+1,L1+L2*S);"'s ";Z;". Zu
g:{7SPACES}";
432 PRINT N$(1,1);":";W(2);"{2SPACES
}";N$(L1+1,L1+1);":";ABS(W(0))
435 FOR I=200 TO 50 STEP -8:SOUND 0,
I,10,10:SOUND 1,I+1,10,10:NEXT I:SOU
ND 0,0,0,0:SOUND 1,0,0,0
440 X=X(S):Y=Y(S)
445 COLOR 2:GOSUB 5000
450 K=STICK(S):IF K+STRIG(S)=16 THEN
 450
455 IF STRIG(S)=0 THEN 500
457 COLOR 3:GOSUB 5000
460 X=X-(K=11)*(X>1)+(K=7)*(X<6)
470 Y=Y-(K=14)*(Y>1)+(K=13)*(Y<6)
480 GOTO 445
500 REM *FELDTEST*
510 K=1-2*S:COLOR 3:GOSUB 5000
520 IF SGN(F(X,Y))=-K THEN POKE V,1:
GOSUB 3900:PRINT "{CODE170/198/197/2
01/206/196/204/201/195/200/197/211/1
60/199/197/194/201/197/212/170}";:GO
SUB 3950:GOTO 430
530 B=F(X,Y)
540 IF ABS(B)>=4 THEN GOSUB 3900:PRI
NT "{CODE170/194/213/210/199/160/201
/211/212/160/211/195/200/207/206/160
/193/213/211/199/197/194/193/213/212
/170}";:GOSUB 3950:GOTO 430
550 B=ABS(F(X,Y)):IF B=1 THEN B=0
600 F(X,Y)=F(X,Y)+K+(ABS(F(X,Y))=0)*
K
620 XX=X:YY=Y:B=ABS(F(X,Y))-1:GOSUB
6000
650 REM *FELDWIRKUNG*
700 RESTORE 3100:EN=0:FOR Q=1 TO 4
705 K=-(2*S-1)
710 READ DX,DY
720 P=F(X+DX,Y+DY):IF P=9 OR SGN(P)=
K THEN 750
730 IF P=0 THEN XX=X+DX:YY=Y+DY:B=0:
GOSUB 6000:F(X+DX,Y+DY)=K:GOTO 750
750 NEXT Q
800 REM *ALL TEST*
805 K=1-S*2:W(0)=0:W(2)=0
810 FOR XX=1 TO 6:FOR YY=1 TO 6
820 A=F(XX,YY):IF SGN(A)<>-K THEN 89
0
830 RESTORE 3100:A=0:FOR Q=1 TO 5
840 READ DX,DY:P=F(XX+DX,YY+DY):IF P
=9 THEN 855
850 A=A+SGN(P)*ABS(ABS(P)-1)
855 NEXT Q
860 IF SGN(A)=K THEN B=0:GOSUB 6000:
F(XX,YY)=K
890 P=SGN(F(XX,YY)):W(P+1)=W(P+1)+P:
NEXT YY:NEXT XX
900 GOTO 420-S*10
990 END
1000 REM *WINNER*
1005 GRAPHICS 0
1007 IF ABS(W(0))=W(2) THEN PRINT "*
** REMIS ***":GOSUB 4000:END
1010 PRINT "{CODE125}*** DER GEWINNE
R IST: ";
1050 FOR I=200 TO 50 STEP -4:FOR Q=0
 TO 3
1060 SOUND Q,I,10,10:NEXT Q:NEXT I
1100 S=0:IF ABS(W(0))>W(2) THEN S=1
1110 PRINT N$(L1*S+1,L1+L2*S);" ! **
**"
1120 PRINT "*** MIT ";W(2);" ZU ";AB
S(W(0));" EROBERTEN FELDERN ***"
2000 END
3000 REM *DATEN*
3100 DATA -1,0,0,-1,1,0,0,1,0,0
3900 POKE V,1:POKE H,2:RETURN
3950 GOSUB 4000:PRINT "{CODE156}";:R
ETURN
4000 FOR I=100 TO 170 STEP 15
4010 FOR W=15 TO 0 STEP -1:SOUND 0,I
+W,10,W:SOUND 1,I-W,10,W:NEXT W:NEXT
 I
4020 RETURN
5000 REM *FELDRAHMEN*
5100 PLOT (X-1)*26,(Y-1)*13:DRAWTO X
*26,(Y-1)*13:DRAWTO X*26,Y*13
5110 DRAWTO (X-1)*26,Y*13:DRAWTO (X-
1)*26,(Y-1)*13:RETURN
6000 REM *BURGEN BAUEN*
6100 COLOR 0
6120 FOR I=(YY-1)*13+1 TO YY*13-1
6130 PLOT (XX-1)*26+1,I:DRAWTO XX*26
-1,I:NEXT I
6140 COLOR S+1:BX=(XX-1)*26:BY=YY*13
-1
6150 ON B+1 GOSUB 6200,6300,6500,660
0
6160 RETURN
6200 REM *FAHNE*
6220 PLOT XX*26-12,YY*13-3:DRAWTO XX
*26-12,YY*13-10:DRAWTO XX*26-9,YY*13
-6
6240 FOR I=20 TO 60 STEP 4:SOUND 0,I
,0,10:NEXT I:SOUND 0,0,0,0
6290 RETURN
6300 REM *BURG*
6330 FOR I=BY-5 TO BY:PLOT BX+3,I:DR
AWTO BX+23,I:NEXT I
```

```
6340 PLOT BX+4,BY-9:PLOT BX+22,BY-9
6350 PLOT BX+3,BY-8:DRAWTO BX+5,BY-8
:PLOT BX+21,BY-8:DRAWTO BX+23,BY-8
6360 PLOT BX+2,BY-7:DRAWTO BX+6,BY-7
:PLOT BX+20,BY-7:DRAWTO BX+24,BY-7
6370 PLOT BX+3,BY-6:DRAWTO BX+5,BY-6
:PLOT BX+21,BY-6:DRAWTO BX+23,BY-6
6380 COLOR Ø:FOR I=BX+6 TO BX+2Ø STE
P 2:PLOT I,BY-5:NEXT I
639Ø PLOT BX+4,BY-7:PLOT BX+22,BY-7
64ØØ PLOT BX+4,BY-5:PLOT BX+22,BY-5
641Ø FOR I=BY-3 TO BY:PLOT BX+12,I:D
RAWTO BX+14,I:NEXT I
6415 IF B>1 THEN 649Ø
642Ø FOR I=1 TO 3:SOUND Ø,121,1Ø,8:F
OR W=1 TO 3Ø:NEXT W:SOUND Ø,Ø,Ø,Ø:NE
XT I
643Ø SOUND Ø,81,1Ø,1Ø:FOR W=1 TO 1ØØ
:NEXT W:SOUND Ø,Ø,Ø,Ø
649Ø RETURN
65ØØ REM *AUFBAU*
651Ø GOSUB 63ØØ:COLOR S+1
652Ø FOR I=BY-9 TO BY-5:PLOT BX+11,I
:DRAWTO BX+15,I:NEXT I
653Ø COLOR Ø:FOR I=BY-9 TO BY-7:PLOT
 BX+12,I:DRAWTO BX+14,I:NEXT I
654Ø COLOR S+1:PLOT BX+13,BY-9:DRAWT
O BX+13,BY-5
655Ø PLOT BX+11,BY-8:DRAWTO BX+14,BY
-8
656Ø IF B>2 THEN 659Ø
657Ø SOUND Ø,81,1Ø,8:FOR I=1 TO 6Ø:N
EXT I:SOUND Ø,Ø,Ø,Ø:SOUND Ø,81,1Ø,8:
FOR I=1 TO 3Ø:NEXT I:SOUND Ø,Ø,Ø,Ø
658Ø SOUND Ø,81,1Ø,8:FOR I=1 TO 9Ø:N
EXT I:SOUND Ø,Ø,Ø,Ø
659Ø RETURN
66ØØ REM *DACH*
661Ø GOSUB 63ØØ:COLOR S+1:GOSUB 65ØØ
662Ø FOR I=Ø TO 3:PLOT BX+13-I,BY-11
+I:DRAWTO BX+13+I,BY-11+I
663Ø NEXT I
664Ø SOUND Ø,81,1Ø,8:FOR I=1 TO 3Ø:N
EXT I:SOUND Ø,Ø,Ø,Ø
665Ø FOR W=Ø TO 3:SOUND W,6Ø+W/2,1Ø,
7:NEXT W:FOR I=1 TO 1ØØ:NEXT I:FOR I
=Ø TO 3:SOUND I,Ø,Ø,Ø:NEXT I
669Ø RETURN

 READY !
```

Töne merken

AI Nr. 40

Beschreibung des Programmes (Atari 600XL/800XL, 2200 Bytes)
Haben Sie schon einmal versucht, ein kleines Lied selbst zu komponieren? Dann wissen Sie, wie schwierig es ist, sich Töne einzuprägen. Spätestens nach fünf Minuten sind sie entflogen. Dieses Programm erleichtert Ihnen das Merken von längeren Tonfolgen. Drücken Sie stets die der blau aufleuchtenden Note zugeordnete Zahlen-Taste.

Stefan Schwarz

```
1Ø REM *TOENE MERKEN*
2Ø REM *FUER CHIP-SPECIAL*
3Ø REM *VON STEFAN SCHWARZ*
9Ø GRAPHICS 18:OPEN #1,4,Ø,"K:"
1ØØ SETCOLOR 4,Ø,7:SETCOLOR Ø,3,2:?
#6;CHR$(125)
1Ø5 SETCOLOR 2,8,2
11Ø POSITION 3,4:? #6;"TOENE MERKEN
!"
12Ø FOR X=1 TO 1ØØØ:NEXT X
13Ø ? #6;CHR$(125)
135 GOSUB 15Ø
14Ø GOTO 158
15Ø POSITION 5,5:? #6;"1 2 3 4"
155 RETURN
158 FOR SN=1 TO 5Ø
16Ø A=INT(RND(Ø)*7Ø)+6Ø
17Ø IF A=121 OR A=96 OR A=81 OR A=6Ø
 THEN 2ØØ
18Ø A=A+1:IF A>13Ø THEN A=6Ø
19Ø GOTO 17Ø
2ØØ SOUND Ø,A,1Ø,8:XYZ=Ø
21Ø GOSUB 23Ø
22Ø GOTO 27Ø
23Ø IF A=121 OR XYZ=121 THEN POSITIO
N 5,5:? #6;CHR$(177):RETURN
24Ø IF A=96 OR XYZ=96 THEN POSITION
7,5:? #6;CHR$(178):RETURN
```

```
250 IF A=81 OR XYZ=81 THEN POSITION
9,5:? #6;CHR$(179):RETURN
260 IF A=6Ø OR XYZ=6Ø THEN POSITION
11,5:? #6;CHR$(18Ø):RETURN
27Ø FOR X=1 TO 4ØØ:NEXT X
28Ø SOUND Ø,Ø,Ø,Ø
29Ø GOSUB 15Ø
3ØØ ABC=ABC-A:ABC=ABS(ABC):G=G+1
31Ø IF G=SN THEN GOTO 34Ø
32Ø GOTO 16Ø
34Ø FOR LP=1 TO SN
35Ø GET #1,TON
355 IF TON<49 OR TON>52 THEN 35Ø
36Ø IF TON=49 THEN XYZ=121
37Ø IF TON=5Ø THEN XYZ=96
38Ø IF TON=51 THEN XYZ=81
39Ø IF TON=52 THEN XYZ=6Ø
4ØØ A=Ø:GOSUB 23Ø
41Ø SOUND Ø,XYZ,1Ø,8
42Ø FOR X=1 TO 15Ø:NEXT X
43Ø SOUND Ø,Ø,Ø,Ø
44Ø GOSUB 15Ø
45Ø CV=CV-XYZ:CV=ABS(CV)
46Ø NEXT LP
47Ø G=Ø:IF ABC=CV THEN 5ØØ
48Ø ? #6;" SIE KONNTEN SICH"
49Ø ? #6;"{2SPACES}";SN-1;" TOENE ME
RKEN "
495 GOTO 54Ø
5ØØ POSITION 1,11:? #6;"richtig"
51Ø FOR X=1 TO 5ØØ:NEXT X
52Ø POSITION 1,11:? #6;"{7SPACES}"
53Ø NEXT SN
54Ø POSITION 5,11:? #6;" ende "
55Ø GOTO 55Ø
56Ø REM ** ENDE ***

  READY !
```

Der siebte Sinn für Zahlen

A I Nr. 20

Beschreibung des Programmes (Atari 600XL/800XL, 890 Bytes)
Nicht nur für Sheriffs, die sich gleich mehrere Autokennzeichen der davonbrausenden Gangsterkolonne merken müssen, lohnt dieses Programm. Es hilft Ihnen Ihr Zahlengedächtnis zu schärfen. Egal ob es sich um die neue Telefonnummer, den geheimen Banksafecode oder das Kennzeichen des Unfallflüchters handelt. Auf R U N geht's los.

Stefan Schwarz

```
1 REM *DER 7.SINN FUER ZAHLEN*
2 REM *FUER CHIP-SPECIAL*
3 REM *VON STEFAN SCHWARZ*
4 SETCOLOR 2,1,1:POKE 755,Ø
5 ? CHR$(125);" DER 7.SINN FUER ZAHL
EN"
1Ø ? "GEBEN SIE JEDE ZIFFER,DIE SIE
"
15 ? "SICH MERKEN KONNTEN,EINZELN"
17 ? "HINTER DEM FRAGEZEICHEN EIN !"
18 ? "ES WERDEN PRO DURCHLAUF NEUE"
19 ? "ZAHLEN GENANNT,JEWEILS EINE ME
HR."
2Ø FOR X=1 TO 65ØØ:NEXT X
3Ø ? CHR$(125):SETCOLOR 2,Ø,Ø
4Ø FOR ZX=1 TO 5Ø
5Ø A=INT(RND(Ø)*1Ø)
6Ø FOR X=Ø TO 39
7Ø ? CHR$(125)
8Ø POSITION X,11:? A
9Ø FOR Z=1 TO 4Ø:NEXT Z
1ØØ NEXT X
1Ø5 ? CHR$(125)
11Ø J=J-A
115 J=ABS(J)
12Ø O=O+1
13Ø IF O=ZX THEN 15Ø
14Ø GOTO 5Ø
15Ø FOR W=1 TO ZX
16Ø INPUT Q
17Ø I=I-Q
175 I=ABS(I)
18Ø NEXT W
185 O=Ø
19Ø IF J=I THEN NEXT ZX
2ØØ ? "SIE HABEN SICH ";ZX-1;" ZAHLE
N "
21Ø ? "{5SPACES}MERKEN KOENNEN"
22Ø END
  READY !
```

Saurer Regen

Beschreibung des Programmes (Atari 600XL/800XL, 6760 Bytes, ein Joystick, für einen Spieler)
Zu den Seltenheiten in der Natur gehören Schnecken mit links herum gewundenen Schneckenhäuschen und Wicken, die sich am Zaun links herum hochwinden. Wissenschaftliche Institute bezahlen für solche Exemplare horrende Preise. Stellen Sie sich vor, Sie sehen bei einem Spaziergang ganz zufällig ein solch rares Exemplar auf einer Wiese stehen. Ihnen fällt auf, daß die Blätter der Pflanze bereits zu welken beginnen. Als Sie sich umschauen, entdecken Sie am Horizont zum Glück die ersten Regenwolken. Die Pflanze steht allerdings etwas ungünstig im Windschatten eines großen Baumes. Sobald die ersten Tropfen fallen, versuchen Sie, das wertvolle Gewächs mit Regenwasser zu versorgen. Doch Vorsicht vor dem saueren Regen. Jeder einzelne Tropfen kann sich wegen der verschmutzten Atmosphäre plötzlich in einen sauren Regentropfen verwandeln. Weichen Sie also dem sauren Regen aus und versuchen Sie nur einwandfreies Wasser aufzufangen. Gesundes Regenwasser ist goldgelb, schlechtes färbt sich dagegen während des Fallens in schmutziges Blau.

Nach Programmstart können Sie zwischen drei Schwierigkeitsstufen wählen. Während des Spiels sollten Sie auf Ihre Energie achten. Geht sie zu Ende, so verlieren Sie eines Ihrer insgesamt drei Lebenszyklen. Falls ein saurer Regentropfen aufgefangen wird, verliert Ihr Bonus 20 Punkte, und Ihre Energie zehn Punkte. Bei jedem aufgefangenen Süßwassertropfen bekommen Sie zehn Punkte aufgeschrieben. Bei jedem nicht aufgefangenen, giftigen Tropfen, erhalten Sie 30 positive Punkte. Bei jedem nicht aufgefangenen süßen Tropfen zieht Ihnen der Computer vier Energiepunkte ab. Nach dem ersten überstandenen Regenguß erblüht Ihre Blume und ruht sich beim Lied des Windes in der Sonne aus. Plötzlich zieht ein Sturm auf. Blitz und Donner kündigen einen noch heftigeren Regenguß an. Nach jeder geschafften Runde erhalten Sie ein Zusatzleben; der Bonus wird Ihnen auf Ihr Punktekonto gutgeschrieben. Negativer Bonus wird vom Punktestand abgezogen. Sollten Sie einen neuen Rekord aufstellen, dann tragen Sie Ihren Namen bitte zusätzlich neben der Rekordsumme ein (max. zehn Buchstaben).

Stefan Schwarz

```
2 REM ** SAURER REGEN **
3 REM *FUER CHIP-SPECIAL*
4 REM *VON STEFAN SCHWARZ*
10 GRAPHICS 2:SETCOLOR 2,0,0
20 POSITION 2,4:? #6;" SaUrEr rEgEn"
25 POSITION 2,7:? #6;"{4SPACES}{CODE
238/239/242/237/225/236}"
30 ? " S T A R T DREUCKEN !!{2SPACES
}ODER":?
32 ? "O P T I O N => *ANDERE STUFE*"
35 IF PEEK(53279)=6 THEN 60
40 IF PEEK(53279)=3 THEN GOSUB 50
45 GOTO 35
50 OPI=OPI+1:IF OPI=3 THEN OPI=0
52 FOR X=1 TO 255 STEP 5:SOUND 0,X,1
0,8:NEXT X:SOUND 0,0,0,0
55 IF OPI=0 THEN POSITION 2,7:? #6;"
{4SPACES}{CODE238/239/242/237/225/23
6}{4SPACES}"
57 IF OPI=1 THEN POSITION 2,7:? #6;"
{4SPACES}{CODE237/233/244/244/229/23
6}{4SPACES}"
58 IF OPI=2 THEN POSITION 2,7:? #6;"
{4SPACES}{CODE243/227/232/247/229/24
2}{4SPACES}"
59 RETURN
60 DIM A(20),B(20),WIN$(10)
62 LE=2:P=39:PU=0
65 IF OPI=0 THEN 100
70 IF OPI=1 THEN ST=2:GOTO 2110
80 IF OPI=2 THEN ST=4:GOTO 2110
100 C=6:I=2.8:EN=45:BO=100:VB=60:TG=
```

```
26:ST=Ø:Z=Ø
300 GRAPHICS 5
310 COLOR 2:PLOT 5,Ø:DRAWTO 74,Ø
320 DRAWTO 74,39:DRAWTO 5,39:DRAWTO
5,Ø
335 GOSUB 1500
340 FOR X=1 TO 20
345 GOSUB 350
346 NEXT X
347 GOTO 500
350 B(X)=INT(RND(Ø)*19+1)
360 A(X)=INT(RND(Ø)*68+6)
365 FOR Y=X-1 TO Ø STEP -1
370 IF A(X)=A(Y) THEN 360
375 NEXT Y
380 COLOR 2:PLOT A(X),B(X)
385 RETURN
400 S=STICK(Ø):K=P:COLOR 1
410 IF S=10 OR S=11 OR S=9 THEN P=P-
I
420 IF S=6 OR S=7 OR S=5 THEN P=P+I
425 IF S=15 OR S=14 OR S=13 THEN 460
430 IF P>72 THEN P=72
440 IF P<7 THEN P=7
450 COLOR Ø:PLOT K-1,35:PLOT K,36:PL
OT K+1,35:COLOR 1
460 PLOT P-1,35:PLOT P,36:PLOT P+1,3
5
470 RETURN
500 GL=INT(RND(Ø)*20+1):PL=INT(RND(Ø
)*C):D=Ø:VB=VB-1:IF VB<21 THEN GL=VB
505 IF VB<1 THEN 2000
510 FOR X=B(GL)+3 TO 39
513 J=2:D=D+2:SOUND Ø,D,10,4
515 IF PL=Ø AND X>TG THEN J=3
517 LOCATE A(GL),X,Z
518 IF Z=1 THEN 600
520 COLOR J:PLOT A(GL),X
530 COLOR Ø:PLOT A(GL),X-3
540 GOSUB 400
550 NEXT X
555 GOSUB 700
557 X=GL:IF VB<21 THEN 570
560 GOSUB 350
570 GOTO 500
600 IF J=3 THEN 650
610 PU=PU+10
620 COLOR Ø:PLOT A(GL),X:DRAWTO A(GL
),X-3
630 GOTO 1000
650 EN=EN-10:BO=BO-20
660 GOTO 1100
700 IF J=3 THEN 750
710 EN=EN-4
720 GOTO 1200
750 PU=PU+30
760 GOTO 1300
1000 FOR V=255 TO 1 STEP -10
1010 SOUND Ø,V,10,10
1020 NEXT V:HI=Ø:GOTO 1500
1100 FOR V=80 TO Ø STEP -5
1110 COLOR V:PLOT A(GL),X:DRAWTO A(G
L),X-3
1120 SOUND Ø,V,6,10
1130 NEXT V
1140 HI=Ø:GOTO 1500
1200 FOR V=1 TO 20
1210 SOUND Ø,240,10,10:NEXT V
1220 HI=Ø:GOTO 1500
1300 FOR V=1 TO 200 STEP 10
1310 SOUND Ø,V,10,10
1320 NEXT V:HI=Ø:GOTO 1500
1500 IF EN<1 THEN 1600
1505 SOUND Ø,Ø,Ø,Ø:? CHR$(125):POKE
755,Ø
1510 ? "{6SPACES}ENERGY:";EN;"{2SPAC
ES}PUNKTE:";PU
1520 ? "{7SPACES}LEBEN:";LE;"{4SPACE
S}BONUS:";BO
1522 ? "{4SPACES}**PUNKTE-REKORD***
";SCORE;" **"
1525 IF EN=Ø THEN Z=Ø
1527 IF HI=1 THEN Z=1
1530 IF Z=1 THEN 557
1540 RETURN
1600 EN=Ø:GOSUB 1505
1610 FOR V=1 TO 150
1620 SO=INT(RND(Ø)*250+2)
1630 SOUND Ø,SO,10,8
1640 NEXT V
1641 LE=LE-1:IF LE<Ø THEN 1660
1642 IF ST=Ø THEN EN=30
1643 IF ST=1 THEN EN=35
1644 IF ST=2 THEN EN=40
1645 IF ST=3 THEN EN=45
1646 IF ST=4 THEN EN=50
1647 IF ST>4 THEN EN=55
1649 HI=1
1650 GOTO 1500
1660 RESTORE 5000
1665 GOTO 2500
1670 READ TON,ZEIT
1680 IF TON=-1 THEN 2500
1690 SOUND Ø,TON,10,8
1700 FOR V=1 TO ZEIT:NEXT V
1710 GOTO 1670
2000 LE=LE+1:PU=PU+BO:BO=Ø:Z=Ø:HI=Ø
2010 POKE 77,Ø:GOSUB 1500
2020 GOSUB 3000
2100 ST=ST+1
2110 IF ST=1 THEN I=2.5:EN=55:BO=120
:VB=70:TG=27:C=6
2120 IF ST=2 THEN I=2.3:EN=60:BO=140
:VB=85:TG=27:C=5
```

```
2130 IF ST=3 THEN I=2.1:EN=75:BO=160
:VB=100:TG=28:C=5
2140 IF ST=4 THEN I=2:EN=90:BO=180:V
B=120:TG=29:C=4
2150 IF ST>4 THEN I=2:EN=100:BO=200:
YU=YU+10:VB=110+YU:TG=30:C=3
2155 IF ST>6 THEN I=1.6
2160 Z=0:GOTO 300
2500 PU=PU+BO
2700 GRAPHICS 2:SETCOLOR 2,0,0
2710 POSITION 7,2:? #6;" G{CODE225}m
E"
2790 POSITION 7,4:? #6;" {CODE239}Ve
R"
2795 SOUND 0,0,0,0
2800 IF WJ=1 THEN POSITION 2,7:? #6;
WP;" ";WIN$
2810 IF PU<WP THEN 2820
2812 ? "{2SPACES}N E U E R{3SPACES}R
 E K O R D !!"
2813 ? "GEBEN SIE IHREN NAMEN EIN:";
:INPUT WIN$
2815 WP=PU:WJ=1:SCORE=WP
2820 POSITION 2,7:? #6;"{17SPACES}"
2830 POSITION 2,7:? #6;WP;" ";WIN$
2840 ? CHR$(125):? "{2SPACES}NEUES S
PIEL ?"
2850 ? " S T A R T{2SPACES}DRUECKEN
!! "
2860 IF PEEK(53279)<>6 THEN POKE 77,
0:GOTO 2860
2890 GOTO 62
3000 GRAPHICS 5:SETCOLOR 1,12,5:SETC
OLOR 4,8,9:R=2:V=1
3010 FOR X=0 TO 79 STEP 2
3020 COLOR 2:PLOT X,39:DRAWTO X,37
3030 NEXT X:GOSUB 3040
3035 GOTO 3200
3040 COLOR R:SO=0
3050 FOR X=36 TO 30 STEP -0.05
3060 PLOT 39,X:SO=SO+1
3070 SOUND 0,SO,10,8
3080 NEXT X:SOUND 0,0,0,0:COLOR V:SO
=255
3085 RESTORE 4000
3090 FOR X=1 TO 140:IF X=20 OR X=40
OR X=60 OR X=80 OR X=100 OR X=120 OR
 X=140 THEN 3100
3095 GOTO 3130
3100 READ AY,AZ:IF AY=-1 THEN 3200
3110 PLOT 39+AY,29-AZ
3120 PLOT 39-AY,29-AZ
3130 SO=SO-1:SOUND 0,SO,10,8
3140 NEXT X
3150 RETURN
3200 RESTORE 4010
3210 READ TON,ZEIT:IF TON=-1 THEN 32
50
3220 SOUND 0,TON,10,8
3230 FOR X=1 TO ZEIT:NEXT X
3240 GOTO 3210
3250 FOR X=3 TO 9 STEP 3
3257 SO=0
3260 FOR V=-20 TO 20
3265 SO=SO+0.5:SETCOLOR 4,8,SO
3270 SOUND 0,ABS(V),8,1+X
3290 NEXT V:NEXT X
3300 GRAPHICS 5+16:SETCOLOR 0,0,8
3310 PLOT 15,0:DRAWTO 18,20
3320 DRAWTO 25,15:DRAWTO 20,39
3330 FOR X=1 TO 255
3340 SOUND 0,X,6,8
3350 SETCOLOR 0,0,X
3360 NEXT X
3370 RETURN
4000 DATA 0,0,1,0,2,1,3,2,3,3,3,4,2,
5,-1,-1
4010 DATA 96,40,91,40,81,80,96,80,0,
5,96,80,0,5,96,80,108,80,0,5,108,40,
96,40,91,80,108,80,0,5,108,40
4020 DATA 72,40,81,80,0,80,96,40,91,
40,81,80,96,80,0,5,96,40,0,5,96,40,0
,5,96,80,108,80,0,5,108,40,96,40
4030 DATA 91,80,128,80,0,5,128,80,12
1,100,0,0,-1,-1

READY !
```

Suchen Sie ein ganz bestimmtes Programm

für Ihre Problemlösung?

Wir bereiten laufend neue CHIP-Specials vor. Wenn Sie uns Ihr Problem und den Computertyp, auf dem es geschrieben werden soll nennen, haben Sie die Chance, es in einem der nächsten CHIP-Specials zu finden.

Über Ihren Vorschlag freut sich:

Die Redaktion CHIP-Special
Armin Schwarz
Bavariaring 8

8000 München 2

X und O

Beschreibung des Programmes (Atari 600 XL/800 XL, 5100 Bytes, 1 oder 2 Joysticks) Für einen oder zwei Spieler.
Wer dieses Spiel schon einmal auf einer beschlagenen Fensterscheibe gespielt hat, braucht die Spielregeln nicht mehr zu lernen.

Um das Spiel zu gewinnen, müssen Sie drei X bzw. drei O waagrecht, senkrecht oder diagonal in eine Reihe setzen. Wenn Sie mehrere Runden mit einem Patner spielen, sollten sich die beiden Spieler beim ersten Zug abwechseln.
Viel Spaß! Stefan Schwarz

```
10 REM * X UND O *
20 REM *FUER CHIP-SPECIAL*
30 REM *VON STEFAN SCHWARZ*
90 OPEN #1,4,0,"K:"
100 DIM X$(1),S1$(12),S2$(12)
110 GRAPHICS 2
115 SETCOLOR 2,0,0
120 POSITION 4,4:? #6;" X {CODE245/2
38/228} O "
122 ? " SPIEL MIT COMPUTER ? = (1)"
123 ? " SPIEL MIT PARTNER{2SPACES}?
= (0)"
124 ? "{2SPACES}WAEHLEN SIE NUN !!!"
125 GET #1,HH:CPL=HH-48:? CHR$(125)
126 IF CPL<>1 AND CPL<>0 THEN 122
130 ? "NAME DES 1.SPIELERS";:INPUT S
1$
135 IF CPL=1 THEN S2$="DES COMPUTER"
:GOTO 150
140 ? "NAME DES 2.SPIELERS";:INPUT S
2$
150 GOTO 500
160 AP=0:AP1=0
163 RESTORE 3050
165 PY=0:PX=0
170 FOR X=1 TO 3
180 READ N,M:IF N=-1 THEN 300
190 LOCATE N,M,PP
200 IF PP=ASC("{CODE216}") THEN PY=P
Y+1
210 IF PP=ASC("o") THEN PX=PX+1
220 IF PP=32 THEN B1=N:B2=M
230 NEXT X
240 IF PY+PX=3 THEN 165
245 IF AP1=1 AND PY=2 THEN 270
250 IF PX=2 THEN 270
260 GOTO 165
270 POSITION B1,B2:? #6;"o"
280 GOSUB 1220:GOTO 900
300 AP=AP+1:IF AP=1 THEN AP1=1
310 IF AP=2 THEN 270
320 GOTO 163
500 GRAPHICS 2
505 SETCOLOR 2,0,4:SETCOLOR 0,8,5
510 FOR X=1 TO 9
520 POSITION 6,X:? #6;"#"
530 POSITION 10,X:? #6;"#"
540 NEXT X
550 FOR X=3 TO 13
560 POSITION X,3:? #6;"#"
570 POSITION X,7:? #6;"#"
580 NEXT X
590 Z=Z+1:IF Z/2=INT(Z/2) THEN TE=1:
GOTO 610
600 TE=0
610 IF TE=0 THEN Z0=Z0+1
620 IF TE=1 THEN Z1=Z1+1
625 ? CHR$(125)
630 IF TE=0 THEN ? " ";S1$;"'S ";Z0;
".ZUG"
640 IF TE=1 THEN ? " ";S2$;"'S ";Z1;
".ZUG"
645 IF CPL=1 AND TE=1 THEN 160
650 ? "{2SPACES}JOYSTICK BEWEGEN ,"
660 ? " UND {CODE203/206/207/208/198
} DRUECKEN !"
670 IF TE=0 THEN RESTORE 3000
680 IF TE=1 THEN RESTORE 3010
690 READ X$
700 REM ******STEUERN********
710 IF STICK(TE)=10 AND STRIG(TE)=0
THEN Y=10:GOSUB 1050:POSITION 4,1:?
#6;X$:GOSUB 1220:GOTO 900
720 IF STICK(TE)=14 AND STRIG(TE)=0
THEN Y=14:GOSUB 1050:POSITION 8,1:?
#6;X$:GOSUB 1220:GOTO 900
730 IF STICK(TE)=6 AND STRIG(TE)=0 T
HEN Y=6:GOSUB 1050:POSITION 12,1:? #
6;X$:GOSUB 1220:GOTO 900
740 IF STICK(TE)=11 AND STRIG(TE)=0
THEN Y=11:GOSUB 1050:POSITION 4,5:?
#6;X$:GOSUB 1220:GOTO 900
750 IF STICK(TE)=15 AND STRIG(TE)=0
THEN Y=15:GOSUB 1050:POSITION 8,5:?
#6;X$:GOSUB 1220:GOTO 900
760 IF STICK(TE)=7 AND STRIG(TE)=0 T
```

```
HEN Y=7:GOSUB 1050:POSITION 12,5:? #
6;X$:GOSUB 1220:GOTO 900
770 IF STICK(TE)=9 AND STRIG(TE)=0 T
HEN Y=9:GOSUB 1050:POSITION 4,9:? #6
;X$:GOSUB 1220:GOTO 900
780 IF STICK(TE)=13 AND STRIG(TE)=0
THEN Y=13:GOSUB 1050:POSITION 8,9:?
#6;X$:GOSUB 1220:GOTO 900
790 IF STICK(TE)=5 AND STRIG(TE)=0 T
HEN Y=5:GOSUB 1050:POSITION 12,9:? #
6;X$:GOSUB 1220:GOTO 900
800 GOTO 700
900 RESTORE 3020
910 READ Q,C
920 IF Q=-1 OR C=-1 THEN 970
930 SOUND 0,Q,10,8
940 FOR X=1 TO C:NEXT X
950 SOUND 0,0,0,0
960 GOTO 910
970 GOTO 590
1050 IF Y=10 THEN LOCATE 4,1,OP:GOTO
 1200
1060 IF Y=14 THEN LOCATE 8,1,OP:GOTO
 1200
1070 IF Y=6 THEN LOCATE 12,1,OP:GOTO
 1200
1080 IF Y=11 THEN LOCATE 4,5,OP:GOTO
 1200
1090 IF Y=15 THEN LOCATE 8,5,OP:GOTO
 1200
1100 IF Y=7 THEN LOCATE 12,5,OP:GOTO
 1200
1110 IF Y=9 THEN LOCATE 4,9,OP:GOTO
1200
1120 IF Y=13 THEN LOCATE 8,9,OP:GOTO
 1200
1130 IF Y=5 THEN LOCATE 12,9,OP:GOTO
 1200
1200 REM
1205 IF TE=0 AND OP=ASC("o") THEN GO
SUB 2000
1210 IF TE=1 AND OP=ASC("{CODE216}")
 THEN GOSUB 2000
1215 RETURN
1220 ZZ=0
1225 RESTORE 3050
1230 READ N,M
1235 IF N=-1 OR M=-1 THEN RETURN
1240 LOCATE N,M,S
1245 IF S=ASC("o") OR S=ASC("{CODE21
6}") THEN ZZ=ZZ+1
1247 IF ZZ=24 THEN 4003
1250 READ N,M
1255 IF N=-1 OR M=-1 THEN RETURN
1260 LOCATE N,M,T
1265 IF T=ASC("o") OR T=ASC("{CODE21
6}") THEN ZZ=ZZ+1
1267 IF ZZ=24 THEN 4003
1270 READ N,M
1275 IF N=-1 OR M=-1 THEN RETURN
1280 LOCATE N,M,R
1285 IF R=ASC("o") OR R=ASC("{CODE21
6}") THEN ZZ=ZZ+1
1287 IF ZZ=24 THEN 4003
1290 IF TE=0 AND S=ASC("{CODE216}")
AND T=ASC("{CODE216}") AND R=ASC("{C
ODE216}") THEN 4000
1300 IF TE=1 AND S=ASC("o") AND T=AS
C("o") AND R=ASC("o") THEN 4010
1310 GOTO 1230
2000 ? CHR$(125);"*** GEGNERISCHES F
ELD *** "
2010 FOR X=1 TO 255 STEP 0.5
2020 SOUND 0,X,10,8
2030 NEXT X
2040 SOUND 0,0,0,0
2050 GOTO 630
3000 DATA {CODE216
3010 DATA o
3020 DATA 81,80,72,40,81,40,91,40,96
,40,108,40,121,80,60,40,0,15,60,40,-
1,-1
3050 DATA 4,1,8,1,12,1,4,5,8,5,12,5,
4,9,8,9,12,9
3060 DATA 4,1,4,5,4,9,8,1,8,5,8,9,12
,1,12,5,12,9
3070 DATA 4,1,8,5,12,9,4,9,8,5,12,1,
-1,-1,-1
4000 GRAPHICS 2
4002 GOTO 4005
4003 POSITION 6,5:? #6;"REMIE"
4004 GOTO 4050
4005 POSITION 3,1:? #6;S1$:GOTO 4020
4010 GRAPHICS 2
4011 IF CPL=1 THEN S2$="DER COMPUTER
"
4012 POSITION 3,1:? #6;S2$
4020 POSITION 5,3:? #6;"HAT"
4030 POSITION 3,5:? #6;"GEWONNEN!"
4050 FOR X=0 TO 255
4060 SOUND 0,X,8,8
4070 NEXT X
4080 SOUND 0,0,0,0
4090 ? CHR$(125)
5000 ? "WOLLT SIE NOCH EINMAL SPIELE
N?"
5010 GET #1,J:IF J=ASC("J") THEN RUN

5020 END

READY !
```

Finanz-Manager

Beschreibung des Programmes (Atari 600XL/800XL, 4300 Bytes)
Brauchen Sie einen neuen Kredit? Möchten Sie Ihr Kapital anlegen? Wollen Sie für eine Rente einzahlen? Können Sie sich das leisten? Was springt dabei für Sie heraus? Die Antwort gibt dieses Programm. Es bringt Ordnung in Ihre Finanzen.
Der FINANZ-MANAGER besteht aus zwei Teilen:

EINFACHE ZINSESRECHNUNG

Grundlage dieses Teiles ist die Formel: K = K * r .
Dabei bedeutet:

K ... Startkapital
K ... Endkapital nach m Monaten
r .... Aufzinsungsfaktor r=1+p/1200
p .... Zinsfuß (Prozent/Jahr)
m .... Spardauer (Monate)

Das Programm ist so aufgebaut, daß von den vier Größen drei eingegeben werden müssen und die resultierende Größe kalkuliert werden kann. Die zu berechnende Größe wird einfach mit Null eingegeben. Beispiel:

```
K  = 1000 Mark
p  =    8 Prozent-Jahr
m  =   60 Monate
-----------------
K  = 1981 Mark
```

RATEN/ANNUITÄTENRECHNUNG

Hier sind die Verhältnisse etwas komplizierter. Basis der Kalkulation ist folgende Zinsesformel über die sogenannte "nachschüssige" Rente:

$$A = \frac{b}{r} * \frac{r - 1}{r - 1}$$

mit

A ..... Monatliche Rate (Annuität)
b ... Kredithöhe
r ... Aufzinsungsfaktor r=1+p/1200
p ... Zinssatz Prozent-Jahr
m ... Anzahl der Monatsraten

Diese Formel wird auch in abgewandelter Form eingesetzt:

$$K = A * \frac{r - 1}{r - 1}$$

mit

K ... Endkapital nach m Monaten
A .... Monatliche Sparrate
m .... Anzahl der Monate
r .... r = 1 + p/1200
p .... Zinssatz in Prozent/Jahr

Erstes BEISPIEL: Für den Kauf eines neuen Autos müssen 22.800 Mark aufgebracht werden. Wie hoch ist die monatliche Rate bei einem Zinssatz von 8,5 Prozent pro Jahr und einer Kreditlaufzeit von vier Jahren? Unser Programm liefert die Antwort: A = 561,98 Mark.
Zweites BEISPIEL: Welcher Endbetrag kann bei einer monatlichen Sparrate von 521 Mark, einem Zinsfuß von 9,1 Prozent und einer Spardauer von sechs Jahren angespart werden? Auch hier gibt unser Programm rasch Auskunft: A = 49.657,47 Mark.

PROGRAMM-AUFBAU

Die Zeilen 100 bis 300 bringen die Hauptwahlliste auf den Bildschirm und verzweigen zu den entsprechenden Unterprogrammen. Zeile 350 bis 810: Enthält die Routinen für die einfache Zinseszinsrechnung. Zeile 950 bis 1720: Unterprogramme für die Annuitätenrechnung.
Zeile 1300 bis 1550: Zinssatz iterativ ermitteln. Dieser Programmteil ist sehr interessant. Aus der Formel A(r -1)-KN(r-1)=0 kann nämlich der Aufzinsungsfaktor r = 1 +p/1200 nicht explizit ermittelt werden. Dieses Programmsegment zur Bestimmung des Zinsfußes ist daher ein sehr schönes Anwendungsbeispiel für den "ILLINOIS"-Algorithmus. Zeile 1300 bis 1320: Definiert automatisch das Intervall, in dem die Nullstelle liegt. Dabei wird zunächst von einem vermuteten Zinssatz p = 8 Prozent --- r = 0.00666666 ausgegangen und das Intervall beidseitig solange vergrößert, bis Funktionswerte gefunden werden, die positiv und negativ sind. Zeile 5000 bis 5110: Checkt die Tastatur und überprüft die gedrückte Taste mit den Optionen KEY1$ und KEY2$. Zeile 6000 bis 6100: Meldet eine falsche Eingabe.

Zeile 7000 bis 7090: Wartet solange, bis die Taste "J" oder "j" gedrückt wurde. In diesem Fall wird ein Flag Q=1 gesetzt. Bei jeder anderen Taste wird Q=0 gesetzt.

J.Peschetz

```
110 DATA ****************************
**
130 DATA **{2SPACES}FINANZ{2SPACES}-
-{2SPACES}MANAGER{4SPACES}**
140 DATA **{2SPACES}FUER CHIP-SPECIA
L{6SPACES}**
150 DATA **{2SPACES}VON J.PESCHETZ{9
SPACES}**
200 DATA ****************************
**
220 DATA ,-- ZINSESZINSRECHNUNG.....
<1>
230 DATA -- RATEN / ANNUITAETEN....<
2>
235 DATA -- Arbeit beenden.........<
3>
240 DATA ,EOF
250 DIM M$(30):OPEN #1,4,0,"K:"
260 PRINT CHR$(125):SETCOLOR 2,8,0
270 RESTORE
280 READ M$:IF M$<>"EOF" THEN PRINT
M$:GOTO 280
290 KEY1=ASC("1"):KEY2=ASC("3"):GOSU
B 5050
300 ON A-48 GOSUB 400,1000,330
310 GOTO 260
320 CLOSE #1
330 END
400 PRINT CHR$(125)
410 PRINT "Z I N S E S Z I N S R E C
 H N U N G"
415 PRINT "=========================
==========":PRINT
520 PTR=0:TRAP 520
530 POSITION 5,8
540 PRINT "{CODE193/206/198/193/206/
199/211/203/193/208/201/212/193/204}
:{2SPACES}";:INPUT K0
545 IF K0=0 THEN PTR=1
550 TRAP 550
560 POSITION 5,10
570 PRINT "{CODE218/201/206/211/211/
193/212/218/186}{8SPACES}";:INPUT P
572 IF P<>0 THEN 580
575 IF PTR=0 THEN PTR=2:GOTO 580
576 GOSUB 6050:GOTO 520
580 TRAP 580
590 POSITION 5,12
600 PRINT "{CODE205/207/206/193/212/
197/186}{10SPACES}";:INPUT M
602 IF M<>0 THEN 610
605 IF PTR=0 THEN PTR=3:GOTO 610
606 GOSUB 6050:GOTO 520
610 TRAP 610
620 POSITION 5,14
630 PRINT "{CODE197/206/196/203/193/
208/201/212/193/204/186}{6SPACES}";:
INPUT KN
632 IF KN<>0 THEN 640
635 IF PTR=0 THEN PTR=4:GOTO 640
636 GOSUB 6050:GOTO 520
640 IF PTR=0 THEN GOSUB 6050:GOTO 52
0
650 ON PTR GOTO 680,710,740,770
680 K0=KN/(1+P/1200)^M
690 POSITION 21,8:PRINT "<>";K0;
700 GOTO 800
710 R=(KN/K0)^(1/M):P=(R-1)*1200
720 POSITION 21,10:PRINT "<>";P;
730 GOTO 800
740 M=LOG(KN/K0)/LOG(1+P/1200)
750 POSITION 21,12:PRINT "<>";M;
760 GOTO 800
770 KN=K0*(1+P/1200)^M
780 POSITION 21,14:PRINT "<>";KN;
785 POSITION 5,18
786 PRINT "ZINSBETRAG:{7SPACES}";KN-
K0
800 GOSUB 7050:IF Q=1 THEN 520
810 RETURN
1000 PRINT CHR$(125)
1005 PRINT "RATEN / ANNUITAETEN - RE
CHNUNG"
1010 PRINT "========================
======":PRINT
1020 PTR=0:TRAP 1020
1030 POSITION 5,8
1040 PRINT "{CODE210/193/212/197/175
/193/206/206/213/201/212/193/197/212
/186}{2SPACES}";:INPUT A
1050 IF A=0 THEN PTR=1
1060 TRAP 1060
1070 POSITION 5,10
1080 PRINT "{CODE218/201/206/211/211
/193/212/218/186}{8SPACES}";:INPUT P
1090 IF P<>0 THEN 1120
1100 IF PTR=0 THEN PTR=2:GOTO 1120
1110 GOSUB 6050:GOTO 1020
1120 TRAP 1120
1130 POSITION 5,12
1140 PRINT "{CODE205/207/206/193/212
/197}{11SPACES}";:INPUT M
1150 IF M<>0 THEN 1175
1160 IF PTR=0 THEN PTR=3:GOTO 1175
```

```
1170 GOSUB 6050:GOTO 1020
1175 IF PTR=0 THEN 1650
1180 TRAP 1180:POSITION 5,14
1182 IF PTR=2 OR PTR=3 THEN PRINT "{
CODE197/206/196/203/193/208/201/212/
193/204/186}{6SPACES}";:GOTO 1187
1185 PRINT "{CODE196/193/210/204/197
/200/197/206/186}{8SPACES}";
1187 INPUT SN
1190 IF SN<>0 THEN 1220
1200 IF PTR=0 THEN PTR=4:GOTO 1220
1210 GOSUB 6050:GOTO 1020
1220 IF PTR=0 THEN GOSUB 6050:GOTO 1
020
1225 TRAP 44444
1230 ON PTR GOTO 1240,1300,1600,1650
1240 R=1+P/1200:A=SN*R^M*(R-1)/(R^M-
1)
1250 POSITION 21,8:PRINT "<>";A
1260 GOTO 1700
1300 X1=1.00666666:X2=1.00666666
1305 X1=X1-1.0E-03:X=X1:GOSUB 1550:F
X1=FX
1310 X2=X2+1.0E-03:X=X2:GOSUB 1550:F
X2=FX
1320 IF FX1*FX2>0 THEN 1305
1330 X3=X2-(X2-X1)/(FX2-FX1)*FX2
1340 X=X3:GOSUB 1550:FX3=FX
1350 IF FX3=0 THEN 1500
1360 IF ABS(FX3)<1.0E-05 THEN 1500
1370 IF FX3*FX2>0 THEN 1410
1380 X1=X2:FX1=FX2
1400 GOTO 1420
1410 FX1=FX1/2
1420 X2=X3:FX2=FX3:GOTO 1330
1500 P=(X3-1)*1200
1510 POSITION 21,10:PRINT "<>";P;
1520 GOTO 1700
1530 REM
1550 FX=A*(X^M-1)-SN*(X-1):RETURN
1560 REM
1600 R=1+P/1200
1610 M=LOG(SN/A*(R-1)+1)/LOG(R)
1620 POSITION 21,12:PRINT "<>";M
1630 GOTO 1700
1650 R=1+P/1200:SN=A*(R^M-1)/(R-1)
1660 POSITION 5,14:PRINT "{CODE197/2
06/196/203/193/208/201/212/193/204/1
86}{5SPACES}<>";SN
1700 POSITION 5,18
1710 GOSUB 7050:IF Q=1 THEN 1020
1720 RETURN
5050 POSITION 2,22
5060 PRINT "{CODE196/197/201/206/197
/160/215/193/200/204/191} ";
5070 GET #1,A
5090 IF A<KEY1 OR A>KEY2 THEN GOSUB
6050:GOTO 5070
5100 POSITION 2,22:PRINT "{11SPACES}
";
5110 RETURN
6050 X=PEEK(85):Y=PEEK(84):POSITION
2,23:PRINT "FALSCHE EINGABE!!!";
6055 SOUND 0,145,10,15
6060 SOUND 1,140,10,13
6070 SOUND 2,217,5,8
6080 FOR I=1 TO 50:NEXT I
6090 SOUND 0,0,0,0:SOUND 1,0,0,0:SOU
ND 2,0,0,0
6095 POSITION 2,23:PRINT "{18SPACES}
";:POSITION X-1,Y:PRINT " ";
6100 RETURN
7050 POSITION 2,23:PRINT "{CODE215/1
97/201/212/197/210/160/168/202/175/2
06/169/191}";
7060 Q=PEEK(764):IF Q=255 THEN 7060
7070 POKE 764,255
7080 IF Q=1 OR Q=106 THEN Q=1:RETURN
7090 Q=0:RETURN
  READY !
```

Sortieren numerischer Felder

Beschreibung des Programmes 600XL/800XL, 2590 Bytes)
Moderne Datenverarbeitungsanlagen sind ohne leistungsfähige Sortierprozesse undenkbar. Für uns ist Sortieren daher eine Gelegenheit, die Leistungsfähigkeit bestimmter Sortieralgorithmen zu untersuchen. Dabei betrachten wir hier das Sortieren von numerischen Feldern.

Die wichtigste Voraussetzung für das Sortieren von Feldern ist der wirtschaftliche Einsatz des Computerspeichers. Das bedeutet, daß die Umstellung der Elemente am Ort zu erfolgen hat. Sortiermethoden, die Elemente eines Feldes Y zu einem Feld X transportieren, sind nicht sehr effizient. Neben dem Kriterium der Optimierung des vorhandenen

Speichers ist auch noch der Zeitaufwand für die Bewertung einer Sortiermethode von entscheidender Bedeutung.
Es gilt allgemein als gutes Maß für die Effizienz, die Anzahl der Schlüsselvergleiche C und die Anzahl M der Umstellungen der Elemente. M und C sind von der Anzahl N der zu sortierenden Elemente abhängig. Im Rahmen unseres Programmes betrachten wir drei Sortiermethoden: BUBBLE SORT, SUPER SHELL SORT und QUICK SORT.

PROGRAMM-AUFBAU
Zeile 100 bis 280: Daten für Hauptwahlliste. Zeile 300 bis 315: Wichtige Felder:
S(NMAX)... zu sortierendes Feld,
M$(30) ... Hilfsstring,
K1(15) .... Indexfelder für
K2(15)...QUICK SORT
STB$(1)... Tabulator aufheben,
TB$(1)... Tabulator setzen.
Zeile 330 bis 840: SORT-TEST. Zeile 380 bis 420: Bringt die Hauptwahlliste und nimmt die Wahl der Sortierroutine (S) und die Anzahl (N) der zu sortierenden Elemente entgegen. Zeile 430: Wählt ganze Zufallszahlen zwischen 1 und 100 und setzt N Stück davon in den Array S(N). Zeile 440: Die unsortierten Felder werden sauber formatiert über den Bildschirm ausgegeben. Zeile 450 bis 550: Durchführung der mit S vorgewählten Sortierroutine. Zeile 470: S = 1 : BUBBLE SORT, Zeile 490: S = 2 : SHELL SORT, Zeile 510: S = 3 : QUICK SORT. Zeile 540 bis 560: Weitersortieren oder Arbeit beenden. Zeile 740 bis 840: Array ausdrucken. Druckt den unsortierten oder sortierten Array S(), achtspaltig am Bildschirm unter Verwendung der Tabulatorfunktion. Zeile 795: Hebt die vorhandenen Tabulatormarken auf. Zeile 796: Setzt sechs neue Tabulatormarken.

BUBBLE SORT
Das schönste an dieser Sortiermethode ist ihr Name: "Bläschen Methode". Dieser Sortieralgorithmus fällt unter die Gruppe der mit "Sortieren durch direktes Austauschen" bezeichneten Methoden. Der Prozeß beruht auf dem Prinzip des fortgesetzten Vergleichens und Vertauschens direkt nebeneinander liegender Paare von Elementen, solange bis alle Elemente sortiert sind. Dabei steigen die kleineren Elemente wie Bläschen nach oben.

0	!	1	2	3	4	5	6	7	8
5	!	5	5	1	1	1	1	1	1
10	!	7	1	5	5	4	4	2	2
7	!	1	7	6	4	5	2	3	3
1	!	8	6	4	6	2	3	4	4
8	!	6	4	7	2	3	5	5	5
6	!	4	8	2	3	6	6	6	6
4	!	9	2	3	7	7	7	7	7
9	!	2	3	8	8	8	8	8	8
2	!	3	9	9	9	9	9	9	9
3	!	10	10	10	10	10	10	10	10

BUBBLE SORT --> Sortierrichtung
Die Zahl der Vergleiche beim "BUBBLE SORT" ist: C =(n -n)/2. Die mittlere Anzahl der Bewegungen ist: M(ittel)+3*(n -n)/4. Das bedeutet, daß bei einem Array von 100 Elementen, die annähernd zufällig verteilt sind: C = (100 -100)/2 = 4950 Vergleiche und M(ittel) = 38(100 - 100)/4 = 7425 Bewegungen von Elementen auszuführen sind. Zeile 10260 bis 10360: BUBBLE SORT. Zeile 10310: N0 = -1 ... Flag für Austauschen. Zeile 10350: Solange N0=0, muß das Vergleichen und Austauschen fortgesetzt werden.

SHELL SORT
Entscheidend für die Langsamkeit des "BUBBLE SORT" ist offenbar die Tatsache, daß nur nebeneinander liegende Elemente verglichen und vertauscht werden. Der Vorschlag, weiter voneinander entfernt liegende Elemente zu betrachten, zu vergleichen und zu vertauschen, machte 1959 D. L. SHELL. Die nach ihm benannte Methode, Sortieren durch direktes Einfügen mit abnehmender Schrittweite, ist beträchtlich schneller. Betrachten wir es an unserem Standardbeispiel:

0	!	1	2	3	4
5	!	1	1	1	1
10	!	10	8	8	2
7	!	7	7	2	3
1	!	3	3	3	4
8	!	8	9	9	5

6 !	6	6	6	6
4 !	4	4	4	4
9 !	9	10	10	8
2 !	2	2	7	9
3 !	5	5	5	10

SHELL SORT --> Sortierrichtung
Zeile 10390 bis 10550:SHELL SORT.
Zeile 10450: Im ersten Lauf werden immer Elemente im Abstand R = INT(R*0.3+0.5) = 3 und im zweiten bis vierten Lauf dann Elemente im Abstand von 1 verglichen.
Der ganze Sortierjob ist in vier Läufen, anstelle in acht wie beim "BUBBLE SORT", gelöst. Das ist eine beträchtliche Reduktion des Aufwandes.

QUICK SORT
Dies ist die bisher schnellste bekannte Sortiermethode für Arrays. Ihre Leistung ist so beeindruckend, daß ihr Erfinder C. A. R. HOARE sie "QUICK SORT" nannte. Das Grundprinzip des "QUICK SORT" resultiert aus der Tatsache, daß Austauschen vorzugsweise über größere Distanzen am effizientesten ist.
So kann man zum Beispiel N Elemente, die in umgekehrter Reihenfolge angegeben sind, genau in N/2 Austauschvorgängen sortieren, wenn man von links und rechts außen beginnend, schrittweise von beiden Seiten nach innen fortschreitet. Das bringt allerdings nur dann etwas, wenn wir von vornherein wissen, daß die Sequenz in umgekehrter Reihenfolge vorliegt. Für die Bildung des "QUICK SORT"-Algorithmus können wir dennoch etwas davon lernen:

- Wir zerlegen einen Array in zwei Teilarrays mit einem Verbindungselement X.
- Hierzu wählen wir ein beliebiges Element X und durchlaufen den Array von beiden Seiten von außen nach innen und vertauschen das jeweils größere und kleinere Element gegeneinander.
- Dieser Prozeß des Durchsuchens und Vertauschens wird solange fortgesetzt, bis man sich irgendwo im Array trifft.
- Als Resultat ist nun der Array in Elemente zerlegt, die kleiner als X sind und in welche, die größer als X sind.

Allerdings ist unsere Aufgabe nicht das Zerlegen eines Arrays in Teilsequenzen, sondern seine Sortierung. Vom Zerlegen ist jedoch nur ein kleiner Schritt zum Sortieren. Wir wenden die ersten beiden Operationen solange auf die Teilsequenzen an, bis jeder Teil nur mehr als ein Element umfaßt.

0 :	5	10	7	1	8	5	4	9	2	3
1 :	1	2	3	4	5	6	8	9	7	10
2 :	1	2	3	4	5	6	7	8	9	10

QUICK SORT V Sortierrichtung
Dicke Bücher sind über den "QUICK SORT" geschrieben worden, und wir können uns hier nicht mit vielen Details aufhalten. Wer sich dafür interessiert, dem raten wir zum Studium von N. WIRTH, "Algorithmen und Datenstrukturen".

SORT	ELEMENTE 10	ELEMENTE 100
BUBBLE	2	195
SHELL	1	27
QUICK	1	22

ZUR TABELLE: VERGLEICH DER METHODEN (SEKUNDEN)
Der "BUBBLE SORT" ist für zufällig verteilte Elemente die mit Abstand schlechteste Sortiermethode. Die Vorzüge von "SHELL" und "QUICK SORT" zeigen sich aber erst bei einer größeren Anzahl von zu sortierenden Elementen. Peschetz

```
10 REM *SORTIEREN*
20 REM *FUER CHIP-SPECIAL*
30 REM *VON J.PESCHETZ*
100 DATA ***************************
110 DATA ***************************
120 DATA **{23SPACES}**
130 DATA **{2SPACES}N U M B E R S O
R T{2SPACES}**
140 DATA **{23SPACES}**
150 DATA **{5SPACES}(TESTPROGRAMM){4
SPACES}**
160 DATA **{23SPACES}**
170 DATA **=======================**
180 DATA **{23SPACES}**
190 DATA ** 19840420 - J.PESCHETZ **
200 DATA **{23SPACES}**
210 DATA ***************************
220 DATA ***************************
230 DATA ,
240 DATA BUBBLE SORT.............(1)
260 DATA SHELL SORT..............(2)
270 DATA QUICK SORT..............(3)
280 DATA ,EOF
290 REM
300 NMAX=100:M=15
310 DIM S(NMAX),M$(30),STB$(1),TB$(1
),K1(15),K2(15)
315 STB$=CHR$(159):TB$=CHR$(127)
370 PRINT CHR$(125):SETCOLOR 2,8,0
375 RESTORE 100
380 READ M$:IF M$<>"EOF" THEN PRINT
M$:GOTO 380
390 PRINT "Ihre Wahl bitte...{2SPACE
S}";:INPUT S:IF S<1 OR S>3 THEN 390
400 PRINT "Anzahl der Zahlen{3SPACES
}";
410 INPUT N
420 IF N<3 OR N>NMAX THEN 400
430 FOR I=1 TO N:S(I)=INT(100*RND(0)
+1):NEXT I
440 M$="UNSORTIERTER ARRAY:":GOSUB 7
70
450 ON S GOTO 460,480,500
460 GOSUB 10310
470 M$="BUBBLE{CODE95}SORT:":GOTO 53
0
480 GOSUB 10440
490 M$="SHELL{CODE95}SORT:":GOTO 530
500 GOSUB 10630
510 M$="QUICK{CODE95}SORT:"
530 GOSUB 770
540 PRINT "Weiter durch Tastendruck!
"
550 A=PEEK(764):IF A=255 THEN 550
560 GOTO 370
770 PRINT :PRINT :PRINT
780 PRINT M$
790 FOR I=1 TO LEN(M$):PRINT "-";:NE
XT I
795 FOR I=1 TO 6:PRINT TB$;CHR$(158)
;:NEXT I:PRINT
797 FOR I=1 TO 6:PRINT "{5SPACES}";S
TB$;:NEXT I:PRINT
799 PRINT STR$(S(1));",";
800 FOR I=2 TO N
810 PRINT TB$;STR$(S(I));",";
820 NEXT I
830 PRINT :PRINT
840 RETURN
10060 REM **{23SPACES}**
10310 NO=-1
10320 FOR J=1 TO N-1
10330 IF S(J)>S(J+1) THEN X=S(J):S(J
)=S(J+1):S(J+1)=X:NO=0
10340 NEXT J
10350 IF NO=0 THEN 10310
10360 RETURN
10440 R=N-1
10450 R=INT(R*0.3+0.5)
10460 FOR K=1 TO R
10465 FOR I=K+R TO N STEP R
10470 X=S(I)
10480 FOR J=I-R TO 1 STEP -R
10490 IF X>S(J) THEN POP :GOTO 10520
10500 S(J+R)=S(J)
10510 NEXT J
10520 S(J+R)=X
10530 NEXT I:NEXT K
10540 IF R>1 THEN 10450
10550 RETURN
10630 N1=1:I=1:N2=N
10640 J1=N1:J2=N2
10650 IF S(J1)<=S(J2) THEN 10710
10660 X=S(J1):S(J1)=S(J2):S(J2)=X
10670 J1=J1+1
10680 IF J1=J2 THEN 10730
10690 IF S(J1)<=S(J2) THEN 10670
10700 X=S(J1):S(J1)=S(J2):S(J2)=X
10710 J2=J2-1
10720 IF J2<>J1 THEN 10650
10730 J2=J2+1
10740 IF J2>=N2 THEN 10770
10750 K1(I)=J2:K2(I)=N2
10760 I=I+1
10770 J1=J1-1
10780 IF N1<J1 THEN N2=J1:GOTO 10640
10790 I=I-1
10800 N1=K1(I):N2=K2(I):IF I>0 THEN
10640
10810 RETURN

READY !
```

Programm-Bibliothek

Beschreibung des Programmes (Atari 600XL/800XL/400/800, Programmrecorder, Druckeranschluß möglich, 10720 Bytes)
Wenn es Ihnen zu mühsam ist, Ihre Programme von Hand zu katalogisieren, so benutzen Sie doch den Computer. Nachdem das Programm eingetippt und gestartet wurde, meldet es sich mit dem Hauptmenü. Die gewünschten Unterroutinen können nun durch Eingeben der entsprechenden Zahl aufgerufen werden.

NEUANLAGE:
Mit Hilfe dieser Routine werden beim Ersteinsatz des Programmes Daten aufgenommen.
ANZEIGE:
Gespeicherte Daten werden aufgelistet. Sollten noch keine Daten im Computer vorhanden sein, so werden diese von Band geladen.
UPDATE:
Lädt die gespeicherten Einträge von Band und gibt die Möglichkeit weiterer Eingaben.
SUCHEN:
Diese Unterroutine enthält ein weiteres Menü, das die Auswahl der Suchbegriffe ermöglicht. Grundsätzlich sucht das Programm nach den eingegebenen Zeichen. Außerdem gibt es die Möglichkeit nach Bandnummer und BASIC-/BOOT-Programmen zu suchen.
UNTERMENÜ:
Hiermit wird das Untermenü 1 aufgerufen. Über dieses Menü kann man in folgenden Routinen verzweigen:
SICHERN:
Diese Routine gibt die im Speicher befindlichen Daten (nicht Programme) auf Band aus.
DRUCKEN:
Die gespeicherten Daten werden auf dem Drucker ausgegeben.
LÖSCHEN
Falsche oder überholte Einträge können entfernt werden.
HAUPTMENÜ:
Rückkehr zum Hauptmenü.
INHALT:
Diese Routine erstellt ein Inhaltsverzeichnis für ein bestimmtes Band und gibt es dann auf das Band aus (mit einem Programm zum Anzeigen des Verzeichnisses, es wird mit LOAD "C geladen).
Mit einigen kleineren Änderungen läßt sich dies Programm auch für andere Sammlungen verwenden. Zum Beispiel: Schallplatten, Videobänder, Bücher usw..

PROGRAMM-AUFBAU
Zeile 60 bis 96: Aufbereitung von Haupt- und Untermenü. Zeile 140 bis 310: Eingabe der neuen Daten (Neuanlage, Update). Zeile 320 bis 395: Anzeige der Einträge auf dem Bildschirm. Zeile 1000 bis 1100: Sortierroutine. Zeile 3000 bis 3200: Sicherungsroutine und Bandausgabe. Zeile 3500 bis 3700: Einlesen der Daten von Band und in den Speicher. Zeile 4000 bis 4160: Suchroutine mit Untermenü. Zeile 4300 bis 4370: Druckausgabe. Zeile 4500 bis 4650: Entfernen von Einträgen. Zeile 5000 bis 5180: Inhaltsverzeichnis erstellen. Roland Grießbach

```
2 REM *PROGRAMM-BIBLIOTHEK*
3 REM *FUER CHIP-SPECIAL*
4 REM *VON ROLAND GRIESSBACH*
11 GOSUB 621Ø
5Ø CLR :DIM B$(38),F$(1),ALT$(15),DA
TUM$(8),NEU$(4),U1$(25),U2$(37),U3$(
2Ø)
55 ZEIL=38:ZEIL1=ZEIL-1:U1$="PROGRAM
MBIBILOTHEK STAND ":U2$="{CODE252/17
Ø/17Ø/17Ø/2Ø8/21Ø/2Ø7/199/21Ø/193/2Ø
5/2Ø5/2Ø6/193/2Ø5/197/17Ø/17Ø/17Ø/25
2/194/2Ø6/21Ø/252/218/193/197/2ØØ/2Ø
4/197/21Ø/252/17Ø/194/197/2Ø5/17Ø}"
56 U3$=" PROGRAMMBIBILOTHEK "
6Ø GRAPHICS 1:? #6;U3$:POSITION 6,3:
? #6;"HAUPTMENU":UM=Ø:IX=Ø
65 POSITION Ø,6:? #6;"1 PGMLISTE NEU
ANLAGE":POSITION Ø,9:? #6;"2 PGMLIST
E ANZEIGE":POSITION Ø,12
7Ø ? #6;"3 PGMLISTE UPDATE":POSITION
```

```
Ø,15:? #6;"4 PROGRAMM SUCHEN":POSIT
ION Ø,18:? #6;"5{2SPACES}->{2SPACES}
UNTERMENU 1{2SPACES}"
72 INPUT HM
73 IF SPEICHER>Ø THEN 76
75 ON HM GOSUB 110,3500,3500,4000,80
76 SPEICHER=1:ON HM GOTO 50,320,110,
4020
80 GRAPHICS 1:? #6;U3$:POSITION 5,3:
? #6;"UNTERMENU 1":IX=Ø
85 POSITION Ø,6:? #6;"1 PGMLISTE SIC
HERN":POSITION Ø,9
90 ? #6;"2 PGMLISTE DRUCKEN":POSITIO
N Ø,12:? #6;"3 PROGRAMM DELETEN":POS
ITION Ø,15:? #6;"4 > HAUPTMENU "
92 POSITION Ø,18:? #6;"5 INVERZ PER
BANDNR "
95 INPUT UM:ON UM GOSUB 3ØØØ,432Ø,45
ØØ,6Ø,5ØØØ
96 SPEICHER=1:ON UM GOTO 6Ø,6Ø,6Ø,6Ø
,6Ø
97 END
98 CLR
11Ø GRAPHICS 2:? #6;"BITTE DATUM IN
FORM":? #6:? #6:? #6;"{CODE176/177/1
74/177/177/174/184/179}{2SPACES}EING
EBEN"
12Ø INPUT DATUM$:GRAPHICS Ø:GRAPHICS
 2
14Ø ? #6:? #6;"ANZAHL DER NEUEN":? #
6:? #6;"PROGRAMME{2SPACES}EINGEBEN";
:INPUT NEU
15Ø IF NEU>1 THEN 181
16Ø GRAPHICS 2:POSITION 1,2:? #6;"BI
TTE EINE ZAHL":? #6:? #6;"{2SPACES}>
 1{3SPACES}EINGEBEN !"
17Ø FOR X=Ø TO 1ØØØ:NEXT X:GRAPHICS
2:GOTO 14Ø
181 NEU1=ALT+NEU
182 TRAP 2ØØ:DIM A$(NEU1*ZEIL+11):SP
EICHER=1
2ØØ GRAPHICS Ø:? :? "PROGRAMMDATEN E
INGEBEN ":? :GOSUB 13ØØ
22Ø FOR I=1 TO ZEIL*NEU:A$(I,I)=" ":
NEXT I
23Ø I=1:FOR J=1 TO NEU
24Ø INPUT B$
25Ø IF LEN(B$)=Ø THEN NEU=J-1:GOTO 2
9Ø
26Ø A$(I,I+ZEIL1)=B$
27Ø I=I+ZEIL
28Ø NEXT J
29Ø GRAPHICS 2:? #6;" ES WERDEN":POS
ITION 5,2:? #6;NEU;:POSITION Ø,5:? #
6;" PROGRAMME":? #6;" UEBERNOMMEN"
3Ø1 ON HM GOTO 31Ø,31Ø,3Ø2
3Ø2 A$(LEN(A$)+1)=INPU$(9)
31Ø GOSUB 1ØØØ:REM EINGABEN WERDEN S
ORTIERT
32Ø I=1:J=1:GOSUB 13ØØ
33Ø FOR K=1 TO NEU1:IF K=8 THEN GOSU
B 2ØØØ
34Ø IF K=15 THEN GOSUB 2ØØØ
345 IF K=22 THEN GOSUB 2ØØØ
35Ø IF K=29 THEN GOSUB 2ØØØ
351 IF K=36 THEN GOSUB 2ØØØ
352 IF K=43 THEN GOSUB 2ØØØ
353 IF K=5Ø THEN GOSUB 2ØØØ
354 IF K=57 THEN GOSUB 2ØØØ
358 IF K=9 THEN POKE 82,1
36Ø ? K;".";A$(I,I+ZEIL1)
37Ø I=I+ZEIL:NEXT K
375 IF UM=3 THEN GOSUB 453Ø
376 ? :? "*****ENDE DER PROGRAMMLIST
E*****":? :POKE 82,2
38Ø IF UM=3 THEN GOSUB 46ØØ
385 ? :? ">RETURN< SETZT DAS PROGRAM
M FORT":IF UM=3 THEN LET A$=D$:NEU1=
LEN(D$)/ZEIL
386 INPUT F$
39Ø TRAP 4ØØØØ
395 GOTO 8Ø
4ØØ REM {CODE2Ø8/21Ø/2Ø1/2Ø6/212/197
/21Ø/2Ø7/213/212/2Ø1/2Ø6/197
43Ø I=1:LPRINT :FOR K=1 TO NEU1
45Ø I=I+ZEIL:NEXT K:END
1ØØØ REM {CODE16Ø/211/2Ø7/21Ø/212/21
Ø/2Ø7/213/212/2Ø1/2Ø6/197/16Ø
1Ø1Ø ZEIL1=ZEIL-1:MAX=ZEIL*(NEU1-1)+
1
1Ø4Ø FOR I=1 TO MAX STEP ZEIL
1Ø5Ø DONE=1
1Ø6Ø FOR K=1 TO MAX-I-ZEIL1 STEP ZEI
L
1Ø7Ø KZEIL1=K+ZEIL1:KZEIL=K+ZEIL:KZE
ILZEIL1=KZEIL+ZEIL1
1Ø8Ø IF A$(K,KZEIL1)<A$(KZEIL,KZEILZ
EIL1) THEN GOTO 111Ø
1Ø9Ø DONE=Ø
11ØØ B$=A$(K,KZEIL1):A$(K,KZEIL1)=A$
(KZEIL,KZEILZEIL1):A$(KZEIL,KZEILZEI
L1)=B$
111Ø NEXT K
112Ø IF DONE THEN RETURN
113Ø NEXT I
114Ø RETURN
13ØØ GRAPHICS Ø:? U1$;DATUM$:?
13Ø5 IF HM=4 THEN ? ;"EINGEGEBENER S
UCHBEGRIFF IST >";S$;"<"
13Ø7 IF J=1 THEN POSITION 3,3
131Ø ? U2$:? :RETURN
2ØØØ IF UM=3 THEN 453Ø
2ØØ5 ? "SOLLEN DIE NAECHSTEN PROGRAM
ME":? "ANGEZEIGT WERDEN{3SPACES}J/N
```

```
?"
2010 TRAP 2020:INPUT F$
2020 ? :? :IF F$="N" THEN GRAPHICS 0
:GOTO 380
2030 GRAPHICS 0:GOSUB 1300:RETURN
3000 GRAPHICS 1:? #6;U3$
3003 POSITION 2,9:? #6;"SICHERUNGSRO
UTINE"
3005 POSITION 0,15:? #6;"1 x RETURN
STARTET":POSITION 4,16:? #6;"DIE ROU
TINE":IF IX=1 THEN RETURN
3010 ? "BITTE DATENCASSETTE IN DEN R
ECORDER":? "LEGEN UND BEREIT MACHEN
!"
3020 NEU2=NEU1+1000:NEU$=STR$(NEU2):
LAENGE=11+(NEU1*ZEIL)
3030 IF SI=1 THEN 3101
3100 SI=1:DIM OUTPUT$(LAENGE)
3101 OUTPUT$(LEN(OUTPUT$)+1)=NEU$
3102 OUTPUT$(LEN(OUTPUT$)+1)=DATUM$
3107 OUTPUT$(LEN(OUTPUT$)+1)=A$
3108 OPEN #1,8,0,"C:PGMLISTE"
3109 PRINT #1,OUTPUT$:CLOSE #1:GOTO
60
3500 GRAPHICS 1:? #6;U3$
3501 IF HM=2 THEN POSITION 4,4:? #6;
"PGM{2SPACES}ANZEIGEN"
3502 IF HM=3 THEN POSITION 5,4:? #6;
"PGM{2SPACES}UPDATE{2SPACES}"
3503 POSITION 3,10:? #6;"{CODE197/20
1/206/204/197/211/197/210/207/213/21
2/201/206/197}"
3505 POSITION 0,15:? #6;"1 x RETURN
STARTET":POSITION 4,16:? #6;"DIE ROU
TINE"
3510 ? "BITTE DATENCASSETTE IN DEN R
ECORDER":? "LEGEN UND BEREIT MACHEN
!"
3600 DIM IN$(2),IM$(2):OPEN #2,4,0,"
C:PGMLISTE"
3610 FOR V=1 TO 14:GET #2,A
3611 IN$=CHR$(A):ALT$(LEN(ALT$)+1)=I
N$:NEXT V
3630 ZWI=VAL(ALT$):ALT=ZWI-1000:LAEN
GE=8+(ALT*ZEIL):DIM INPU$(LAENGE)
3640 FOR Z=1 TO LAENGE:GET #2,B
3650 IM$=CHR$(B):INPU$(LEN(INPU$)+1)
=IM$:NEXT Z
3655 CLOSE #2:IF HM=3 THEN RETURN
3660 DIM A$(ALT*ZEIL):LET DATUM$=INP
U$(1,8):LET A$=INPU$(9):NEU1=ALT:RET
URN
4000 REM PGM SUCHEN
4010 GRAPHICS 0:GRAPHICS 1:? #6;U3$:
POSITION 2,4:? #6;"PROGRAMM{2SPACES}
SUCHEN":GOSUB 3502
4020 I=1:IF SX=1 THEN GOTO 4040
4030 DIM S$(25),SU$(1),T$(NEU1*ZEIL)
:SX=1
4040 IF IX=1 THEN SU=2:GOTO 4112
4050 GRAPHICS 1:? #6;" PROGRAMMBIBIL
OTHEK ":POSITION 5,3:? #6;"UNTERMENU
 4":POSITION 0,6:? #6;"SUCHBEGRIFFE:
"
4060 POSITION 0,9:? #6;"1 PGMNAMEN "
:POSITION 0,12:? #6;"2 PGM'S PER BAN
DNR.":POSITION 0,15
4070 ? #6;"3 BASIC/BOOT PGM'S":POSIT
ION 0,18:? #6;"4 {CODE190/160/160/21
2/207/160/200/193/213/208/212/205/19
7/206/213} ":LET T$="":INPUT SU
4101 IX=0:IF SU=4 THEN POP :GOTO 60
4102 I=1:ON SU GOTO 4103,4110,4105
4103 S1=1:GOTO 4110
4105 S1=32:GOTO 4110
4110 GRAPHICS 2:? #6:? #6;"{7SPACES}
BITTE":? #6:? #6;"{3SPACES}SUCHARGUM
ENT",,,:? #6;"{5SPACES}EINGEBEN"
4112 IF IX=1 THEN GRAPHICS 2:? #6:?
#6;"{7SPACES}BITTE":? #6:? #6;"{4SPA
CES}BANDNUMMER ",,,:? #6;"{5SPACES}E
INGEBEN"
4115 IF SU=2 THEN ? "EINGABE 3-STELL
IG z.B.: {CODE176/176/181} "
4117 INPUT S$
4120 GRAPHICS 0:J=1:L=1:GOSUB 1300
4122 FOR K=1 TO NEU1:LET B$=A$(I,I+Z
EIL1)
4124 IF K=10 THEN POKE 82,1:?
4125 IF SU=2 THEN 4150
4133 I=I+ZEIL:IF B$(S1,S1+LEN(S$)-1)
=S$ THEN ? K;".";B$:L=L+1:T$(LEN(T$)
+1)=B$
4134 IF L=8 THEN GOSUB 4300
4135 NEXT K:IF LEN(T$)=0 THEN ? :? :
? "MIT SUCHBEGRIFF ## ";S$;" ## ":?
"NICHTS GEFUNDEN ! "
4139 IF IX=1 THEN RETURN
4140 IF K=NEU1+1 THEN GOSUB 4300
4145 ? :? :? "NOCH EIN SUCHBEGRIFF ?
 (J/N)":POKE 82,2:INPUT F$
4147 IF F$="J" THEN 4050
4149 GOTO 60
4150 I=I+ZEIL:IF B$(20,22)=S$ THEN ?
 K;".";B$:L=L+1:T$(LEN(T$)+1)=B$
4155 IF IX=1 THEN 4135
4160 GOTO 4134
4300 POSITION 2,20:? "SOLLEN DIE ANG
EZEIGTEN PROGRAMME":? "GEDRUCKT WERD
EN ? (J/N)":INPUT F$
4310 L=1:IF F$="N" THEN LET T$="":GR
APHICS 0:RETURN
4320 REM {CODE208/210/201/206/212/19
7/210/207/213/212/201/206/197}
```

```
4325 GRAPHICS 1:? #6;U3$:POSITION 4,
8:? #6;"DRUCKAUSGABE"
4335 LPRINT :LPRINT U1$;DATUM$:V=1
4340 IF HM=4 THEN 4360
4350 FOR X=1 TO (LEN(A$)/ZEIL):LPRIN
T A$(V,V+ZEIL1):GOTO 4370
4360 FOR X=1 TO (LEN(T$)/ZEIL):LPRIN
T T$(V,V+ZEIL1)
4370 V=V+ZEIL:NEXT X:GRAPHICS 0:RETU
RN
4500 REM PGM DELETEN
4510 GRAPHICS 1:? #6;"PROGRAMM-BIBIL
OTHEK":POSITION 2,4:? #6;"PROGRAMM D
ELETEN"
4512 IF SPEICHER>0 THEN 4517
4515 IF HM=5 THEN GOSUB 3500
4517 IF DEL=1 THEN N=0:R=0:DD=0:D$="
":GOTO 4525
4520 DIM D$(NEU1*ZEIL):DIM Z$(4)
4525 ZEIL2=1:DEL=1:GOSUB 320
4530 POSITION 2,20:? "EINTRAG ENTFER
NEN{5SPACES}ENTER : {CODE160/177/173
/185/185/185/160} ";"WEITER IN DER A
NZEIGE ENTER : {CODE160/160/160/176/
160/160/160}":INPUT D
4540 N=D+100:Z$=STR$(N):IF D=0 THEN
2030
4545 IF D>1 THEN 4555
4546 R=D:D=D+1
4550 ZEIL2=ZEIL+1:GOTO 4560
4555 IF D-R=1 THEN ZEIL2=ZEIL2+ZEIL:
R=D:GOTO 4560
4557 D$(LEN(D$)+1)=A$(ZEIL2,(D-1)*ZE
IL):ZEIL2=(D-1)*ZEIL+ZEIL+1:R=D
4560 DD=DD+1:UM1=1:FOR Y=3 TO 23:LOC
ATE 2,Y,V:IF V=ASC(Z$(3)) THEN POSIT
ION 2,Y:? "************OK"
4565 NEXT Y:GOTO 4530
4600 IF ZEIL2>(NEU1*ZEIL) THEN ZEIL2
=ZEIL2-1
4605 IF ZEIL2>(NEU1*ZEIL) THEN ZEIL2
=ZEIL2-1
4610 D$(LEN(D$)+1)=A$(ZEIL2,NEU1*ZEI
L):RETURN
5000 REM {CODE160/201/206/200/193/20
4/212/211/214/197/210/218/174/160/19
8/213/197/210/160/194/193/206/196/16
0/193/213/211/199/174/160}
5010 IX=1:IF SPEICHER=0 THEN GOSUB 3
500
5015 GOSUB 4020
5020 ? "{CODE160/211/207/204/204/197
/206/160/196/201/197/211/197/160/196
/193/212/197/206/160/193/204/211/160
/201/206/200/193/204/212/211/214/197
/210/218/174}":? "{CODE193/213/211/1
99/197/199/197/194/197/206/160/215/1
97/210/196/197/206/160/191/160/202/1
75/206/160}":INPUT F$
5030 IF F$="N" THEN T$="":RETURN
5040 GOSUB 3000
5050 ? "BITTE CASSETTE ";S$;" IN DEN
 RECORDER":? "LEGEN UND BEREIT MACHE
N !"
5060 LIST "C",5100,5180
5065 NEU2=NEU1+1000:OPEN #1,9,0,"C:I
NHALTSVERZ."
5070 ? #1,NEU2;DATUM$;S$;T$
5080 CLOSE #1:T$="":RETURN
5090 GRAPHICS 0:END
5100 REM {CODE170/170/208/199/205/17
3/194/201/194/201/204/207/212/200/19
7/203/160/201/206/200/193/204/212/21
1/214/197/210/218/174/160/242/174/23
1/242/233/229/243/243/226/225/227/23
2/160/170/170}
5120 GRAPHICS 0:DIM ALT$(14),IN$(2),
IM$(3),D$(8):OPEN #2,4,0,"C;INHALTSV
ERZ"
5125 FOR V=1 TO 14:GET #2,A
5130 IN$=CHR$(A):ALT$(LEN(ALT$)+1)=I
N$:NEXT V
5135 ZWI=VAL(ALT$):ALT=ZWI-1000:LAEN
GE=11+(ALT*38):DIM INPU$(LAENGE)
5140 FOR Z=1 TO LAENGE:GET #2,B
5145 IM$=CHR$(B):INPU$(LEN(INPU$)+1)
=IM$:NEXT Z
5150 CLOSE #2:DIM A$(ALT*38):LET D$=
INPU$(1,8):LET A$=INPU$(12):LET IM$=
INPU$(9,11)
5160 ? :? "INHALTSVERZEICHNIS BAND #
 ";IM$;" #":? :? "STAND ";D$:? :?
5170 ? "{CODE170/170/170/208/210/207
/199/210/193/205/205/206/193/205/197
/170/170/170/252/194/206/210/252/218
/193/197/200/204/197/210/252/170/194
/197/205/170}":? :? :? A$
5180 END
6000 ?
6010 ? "{12SPACES}{CODE149/149/149/1
49/149/149/149/149/149/149/149}{2SPA
CES}"
6020 ? "{12SPACES}{CODE160}{3SPACES}
(C){3SPACES}{CODE160}{2SPACES}"
6030 ? "{12SPACES}{CODE160/160/160/1
60/160/160/160/160/160/160/160}{2SPA
CES}"
6040 ?
6050 ? "{12SPACES}{CODE160/160/160/1
60/160} {CODE160/160/160/160/160}{2S
PACES}"
6060 ? "{12SPACES}{CODE160}{3SPACES}
{CODE160} {CODE160}{3SPACES}{CODE160
}{2SPACES}"
```

```
6070 ? "{2SPACES}{CODE17/18/18/18/18
/18/5}{3SPACES}{CODE160}{3SPACES}{CO
DE160} {CODE160}{3SPACES}{CODE160}{3
SPACES}{CODE17/18/18/18/18/18/5}"
6080 ? "{2SPACES}{CODE124/17/23/18/2
3/5/124}{3SPACES}{CODE160/160/160/16
0/160} {CODE160}{7SPACES}{CODE124/17
/23/18/23/5/124}"
6090 ? "{2SPACES}{CODE124/26/24/18/2
4/3/124}{3SPACES}{CODE160/160}{4SPAC
ES}{CODE160} {CODE160/160/160}{3SPAC
ES}{CODE124/26/24/18/24/3/124}"
6100 ? "{2SPACES}{CODE124/8/174/160/
174/10/124}{3SPACES}{CODE160} {CODE1
60}{3SPACES}{CODE160}{3SPACES}{CODE1
60}{3SPACES}{CODE124/8/174/160/174/1
0/124}"
6120 ? "{2SPACES}{CODE26/160/160/160
/160/160/3}{3SPACES}{CODE160}{2SPACE
S}{CODE160}{2SPACES}{CODE160}{3SPACE
S}{CODE160}{3SPACES}{CODE26/160/160/
160/160/160/3}"
6130 ? "{12SPACES}{CODE160}{3SPACES}
{CODE160} {CODE160/160/160/160/160}{
2SPACES}"
6140 ?
6150 ? "{11SPACES}{CODE160/160/160/1
60/160/160/160/160/160/160/160/160/1
60} "
6160 ? "{11SPACES}{CODE160} 22/03/84
{2SPACES}{CODE160} "
6170 ? "{11SPACES}{CODE21/21/21/21/2
1/21/21/21/21/21/21/21/21} "
6189 POSITION 8,21:? #6;"PRESS START
 TO BEGINN"
6190 IF PEEK(53279)=6 THEN RETURN
6205 GOTO 6190
6210 GRAPHICS 0:POKE 752,1:POKE 710,
0:POKE 709,14:POKE 712,146
6220 I=PEEK(560)+PEEK(561)*256+6:POK
E I-3,70:FOR J=0 TO 2:POKE J+I,6:NEX
T J
6230 POSITION 3,0:PRINT #6;"PRORAMMB
IBLOTHEK":POSITION 3,1:PRINT #6;"{2S
PACES}BAND VERSION"
6240 GOTO 6000

  READY !
```

Versandaufkleber

Beschreibung des Programmes (Atari 600XL/800XL/400/800, Drucker, 3270 Bytes)
Dieses Programm gibt Ihnen die Möglichkeit, Ihre Versandaufkleber mit Absender, Empfänger und Versendungsart ausdrucken zu lassen. Sobald Sie das Listing abgetippt haben, wählen Sie per Tastendruck die Versendungsart.
Danach geben Sie die Empfängeranschrift ein. Hat die Postleitzahl mehr als vier Ziffern, so fragt der Computer nach dem Empfängerland. Wollen Sie in irgendeiner Zeile nichts stehen haben, dann drücken Sie lediglich RETURN.
Vergessen Sie nicht in den Zeilen 300 bis 320 Ihre eigene Absenderadresse einzugeben. Erscheint das Aufkleberformular, so brauchen Sie nur "J" zu drücken; Sie bekommen dann einen Ausdruck des Versandaufklebers. Die Druckroutine wurde für einen Matrixdrucker vom Typ Gemini 10X geschrieben. Die Drucksteuerzeichen lassen sich jedoch leicht an andere Drucker anpassen. Achten Sie auf die REMs in der Druckerroutine.
Bernd Moser

```
100 REM *VERSANDAUFKLEBER*
101 REM *FUER CHIP-SPECIAL*
102 REM *VON BERND MOSER **
180 ? CHR$(125)
190 OPEN #1,4,0,"K:"
200 POKE 710,40:POKE 709,3
210 DIM A$(10),B$(15),C$(20),D$(20),
R$(60),E$(20)
220 DIM F$(20),G1$(20),G2$(20),H$(30
),I$(10),J$(30),K$(30)
290 A$="ABS.: "
300 B$="VOGEL-VERLAG KG"
310 C$="BAVARIARING 8"
320 D$="D-8000 MUENCHEN"
```

```
330 FOR A=1 TO 60:R$(A,A)=CHR$(45):N
EXT A
340 ? :? "{6SPACES}VERSENDUNGSART:":
? "{6SPACES}---------------":?
350 ? "{6SPACES}BRIEF..............<
B>"
360 ? "{6SPACES}DRUCKSACHE.........<
D>"
370 ? "{6SPACES}WARENSENDUNG.......<
W>"
380 ? "{6SPACES}PAECKCHEN..........<
P>"
390 ? "{6SPACES}SONSTIGES..........<
S>"
400 ? "{6SPACES}{CODE197/206/196/197
/160/196/197/211/160/208/210/207/199
/210/193/205/205/211/174/188/197/190
}"
410 ? :? "{6SPACES}";
420 GET #1,A:IF A<>ASC("B") AND A<>A
SC("D") AND A<>ASC("W") AND A<>ASC("
P") AND A<>ASC("S") AND A<>ASC("E")
THEN RUN
430 IF A=ASC("S") THEN GOSUB 870
440 IF A=ASC("E") THEN CLOSE #1:? CH
R$(125):END
450 IF A=ASC("B") THEN E$="BRIEF"
460 IF A=ASC("D") THEN E$="DRUCKSACH
E"
470 IF A=ASC("W") THEN E$="WARENSEND
UNG"
480 IF A=ASC("P") THEN E$="PAECKCHEN
"
490 ? CHR$(125):? "SIE WAEHLTEN ";E$
;".":? :?
500 ? "{3SPACES}EMPFAENGER:":? "{3SP
ACES}-----------":?
510 ? "ANREDE......";:INPUT F$:?
520 ? "VORNAME.....";:INPUT G1$:?
530 ? "NACHNAME....";:INPUT G2$:?
540 ? "STRASSE/NR..";:INPUT H$:?
550 ? "PLZ (4-7)...";:INPUT I$:II=LE
N(I$):?
560 ? "WOHNORT.....";:INPUT J$:?
570 IF II>4 THEN ? "EMPFAENGERLAND..
.";:INPUT K$
580 REM
640 ? CHR$(125)
650 POSITION 1,1:? CHR$(17);:FOR A=1
 TO 36:? CHR$(18);:NEXT A:? CHR$(5)
660 FOR I=2 TO 4:POSITION 38,I:? CHR
$(124):NEXT I
670 POSITION 1,2:? CHR$(124);A$;B$:P
OSITION 1,3:? CHR$(124);"{6SPACES}";
C$:POSITION 1,4:? CHR$(124);"{6SPACE
S}";D$
680 POSITION 1,5:? CHR$(1);:FOR A=1
TO 36:? CHR$(18);:NEXT A:? CHR$(4)
690 FOR I=6 TO 20:POSITION 38,I:? CH
R$(124):NEXT I
700 POSITION 1,6:? CHR$(124):POSITIO
N 1,7:? CHR$(124)
710 POSITION 1,8:? CHR$(124),"{6SPAC
ES}";E$
720 POSITION 1,9:? CHR$(124):POSITIO
N 1,10:? CHR$(124)
730 POSITION 1,11:? CHR$(124);"{15SP
ACES}";F$
740 POSITION 1,12:? CHR$(124)
750 POSITION 1,13:? CHR$(124);"{15SP
ACES}";G1$;" ";G2$
760 POSITION 1,14:? CHR$(124);"{15SP
ACES}";H$
770 POSITION 1,15:? CHR$(124)
780 POSITION 1,16:? CHR$(124);"{6SPA
CES}";I$:POSITION 17,16:? J$
790 POSITION 1,17:? CHR$(124)
800 POSITION 1,18:? CHR$(124);"{15SP
ACES}";K$
810 POSITION 1,19:? CHR$(124)
820 POSITION 1,20:? CHR$(26);:FOR A=
1 TO 36:? CHR$(18);:NEXT A:? CHR$(3)
830 POSITION 4,21:? "DRUCKEN........
...<J/N>";:GET #1,A:IF A<>ASC("J") A
ND A<>ASC("N") THEN ? CHR$(128):GOTO
 830
840 IF A=ASC("N") THEN RUN
850 GOSUB 990:REM DRUCKROUTINE
860 GOTO 580
870 ? CHR$(125):? :?
880 ? "WELCHE BESONDERE VERSENDUNGSA
RT:":? :?
890 INPUT E$
900 RETURN
990 LPRINT CHR$(27);CHR$(52);CHR$(27
);CHR$(69)
1000 LPRINT R$
1010 LPRINT "{4SPACES}";A$;B$
1020 LPRINT "{10SPACES}";C$
1030 LPRINT "{10SPACES}";D$
1040 LPRINT R$
1050 LPRINT
1060 LPRINT "{35SPACES}";E$:LPRINT
1070 LPRINT "{35SPACES}";F$
1080 LPRINT "{35SPACES}";G1$;" ";G2$
1090 LPRINT "{35SPACES}";H$
1110 LPRINT CHR$(27);CHR$(87);CHR$(1)
1120 LPRINT "{12SPACES}";I$;" ";J$:L
PRINT
1140 LPRINT CHR$(27);CHR$(87);CHR$(0
)
1150 LPRINT "{35SPACES}";K$
1160 LPRINT R$
1170 GOTO 580
```

READY !

Datenbank

Beschreibung des Programmes (Atari 800XL, Druckeranschluß möglich, 8050 Bytes)
Hinter dem Wort "Datenbank" verbirgt sich ein Dateiprogramm. Wir bringen eine Datenbank für Adressen. Dieses BASIC-Programm kann beliebig abgewandelt als Datenbank für jede Kartei herhalten, zum Beispiel für ein Bücherverzeichnis oder für eine Kochrezeptsammlung usw. Hierzu müssen nur wenige Parameter im Programm verändert werden.
Ein wesentliches Kriterium des Programmes ist, daß die gesamte Kartei auf einmal in den Speicher passen muß, da sie in Form eines überlangen Strings angelegt ist. Das ergibt natürlich eine gewisse Beschränkung hinsichtlich des Umfanges der Kartei, aber für die meisten privaten Anwendungen reicht es.
Das Programm ist bildschirmgeführt, benutzerfreundlich und ermöglicht eine Vielzahl von Funktionen, wie: Adressen eingeben, Adressen suchen, Adressen löschen, Adressen speichern, Adressen ausdrucken, Adressen sortieren.
Auf die Adressen kann über jeden beliebigen Schlüssel in Sekundenbruchteilen zugegriffen werden, ganz wie bei einem Dateiprogramm für die "großen Brüder" des Atari. Wir unterteilen im folgenden die Programmbeschreibung in mehrere Punkte, entsprechend der Wichtigkeit der Funktionen. Das gesamte Programm ist im Abschluß unter dem Beispiel Datenbank angeführt.

BENUTZERFÜHRUNG UND ANWENDERFREUNDLICHKEIT Bei einer großen Zahl - auch von Profianwendern erstellten Programmen - hapert es gewaltig mit der Anwenderfreundlichkeit der Programme. Wir finden, wenn für die Anwendung eines Programmes dicke Bedienungsanleitungen gewälzt werden müssen, ist das Programm schlecht und in der Realität zum Scheitern verurteilt! Der Anwender muß Schritt für Schritt mittels Bildschirmdialoge durch die Funktionen des Programmes geführt werden.
Sobald das Programm gestartet wird, erscheint unmittelbar darauf das Hauptwahlmenü auf dem Bildschirm. Damit ist sofort klar, was als nächstes getan werden kann. Wählen wir die Funktion: ADRESSEN eingeben, erscheint die Bildschirmmaske und wir geben einige Adressen ein. Dann haben wir die Wahl zwischen: K(orrektur)? W(eiter)? F(ertig)?

PROGRAMM-AUFBAU
Zeile 100 bis 280: Daten für das "Adressen Menü".
Zeile 300: Wichtige Deklarationen
S$(NM)...Adressenkartei
SL(0)...Ende Satz 0
SL(1)...Name
SL(2)...Vorname
SL(3)...Straße
SL(4)...Postleitzahl
SL(5)...Stadt
SL(6)...Telefonnummer
Dies ist der Adressen-Record, dargestellt in ATARI-BASIC. Für die Benutzung des Programmes mit ATARI 600XL muß die Dimension des Strings S$(NM) durch Probieren solange reduziert werden, bis das Ganze in den Speicher paßt!
SAN.........Satzanzahl
S...........Schlüssel 1..6
M$,X$.......Hilfsstring
OPT$........Hilfsstring
OPT$-OPT4$..Hilfsstring
N,M,I,J.....Hilfsvariable
K,Z,T,T1....Hilfsvariable
SAL.........Recordlänge
NMAX........Recordanzahl
NM..........Maximale Anzahl der Buchstabenadreßkartei

Zeile 340 bis 440: Programm "ADRESSEN"
Zeile 380: Ein leerer Datensatz sitzt zu Beginn (0..SAL-1) von S$() und enthält in allen Elementen "?"
Zeile 390 bis 400: Bringt das Menü auf den Bildschirm.
Zeile 410 bis 420: Steuert den Ablauf des gesamten Programmes.

Zeile 412 bis 413: Taste "C" ändert die Farbe der Schrift.
Zeile 415 bis 446: Taste "H" ändert die Helligkeit des Hintergrundes der Schrift.
Zeile 418 bis 419: Taste "V" ändert die Helligkeit des Vordergrundes der Schrift.
Es mag überraschen, daß das Hauptprogramm von "ADRESSEN" nur so wenige exekutierbare Programmzeilen mit Anweisungen enthält. Es ist aber im Prinzip der Leistungsfähigkeit einer wohlstrukturierten Programmierung begründet. - Das Hauptprogramm steuert nur in groben Zügen den Ablauf und das Zusammenwirken des gesamten Programmflußes, die eigentliche Arbeit verrichten kurze, überschaubare Funktionen und Prozeduren. So behält man leicht die Übersicht über ein komplexes Programm.

ADRESSEN EINGEBEN

Das Eingeben von Adressen erfolgt über die Bildschirmmaske. Im Programm sieht das folgendermaßen aus:
Zeile 520 bis 610: Adressen eingeben.
Zeile 550: Generiert die Bildschirmmaske.
Zeile 560: Die Laufvariable I für die Datensätze muß beim Eintritt auf die Nummer N des bisher höchsten Datensatzes gesetzt werden. Sobald eine Adresse eingegeben wird, muß der Schlüssel S, der sagt, nach welchem Satzelement die Kartei sortiert ist, auf 0 gesetzt werden; mit jeder Adresseneingabe wird die Sortierung der Datei zerstört! Zeile 570 bis 580: Eingabe der Adressen
Zeile 590 bis 600: Weitere Funktionen
Zeile 610: Nach Abschluß der Eingabe wird die momentane Anzahl der "Karteiblätter" festgelegt.

ADRESSEN SUCHEN

Wie findet man schnell einen Datensatz in einer in Arrayform angelegten Datei, ohne auf ein Element nach dem anderen zugreifen zu müssen? Die Datei in Form eines binären Baumes anzulegen erfordert einen zusätzlichen zweidimensionalen Array für das Festhalten der Schlüssel und würde unseren ohnehin knappen Speicherplatz weiter reduzieren. Ein brauchbarer Algorithmus für unser Problem ist unter dem Namen "High Low" Methode bekannt. Ist die Datei nach aufsteigenden Größenordnungen alphabetisch sortiert, können wir folgenden Algorithmus benutzen:

- o Wir halbieren den Array in der Mitte und vergleichen das mittlere Element mit unserem Suchelement.
- o Ist das Suchelement kleiner als das Dateielement, halbieren wir den linken Teilarray weiter, ist es größer, halbieren wir den rechten Teilarray weiter, nehmen das Mittelelement und vergleichen es mit unserem Suchelement.
- o Ist das Suchelement nicht gleich wie das Dateielement, so setzen wir fort mit dem zuvor genannten Punkt und zwar solange, bis wir das Element gefunden haben oder an den Grenzen des Arrays angekommen sind.

Zeile 640 bis 1380: Adressen suchen.
Zeile 670: Für das Suchen verwenden wir die auf dem Bildschirm dargestellte Maske. Auf die Datei kann folgendermaßen zugegriffen werden: Satznummer, Kriterium (Schlüssel), kompletter Name, Wild-Card, Vorwärtsblättern, Rückwärtsblättern.
Zeile 770 bis 820: Adresse über Satznummer auswählen. Dies ist der einfache Zugriff über den Feldindex.
Zeile 800: In T1 wird die alte ausgewählte Funktion gespeichert, damit später bei der Wahl von "Weiter" mit ihr fortgefahren werden kann.
Zeile 800 bis 810: Eingabe der Satznummer und Ausgabe der Adresse.
Zeile 840 bis 970: Adresse über Kriterium auswählen. Als Kriterium wird der Schlüssel bezeichnet, nachdem die Datei sortiert ist.

S=0 ... Datei ist unsortiert
S=1 ... Datei ist nach Namen sortiert
S=2 ... Datei ist nach Vornamen sortiert
S=3-6.. usw.

Zeile 890: Hat sich aus irgendeinem Grund der Schlüssel S verändert, muß die Datei vor dem Zugriff auf einzelne Blätter neu sortiert werden. Dies besorgt das Subprogramm

SHELL SORT. Manche werden sich wundern, warum wir nicht den schnellen QUICKSORT in dieses Programm eingebaut haben. Der Grund ist, daß eine Kartei fast immer bereits nach einem Schlüssel ganz oder teilweise sortiert vorliegt, da ja meist nur neue Daten hinzugefügt werden. Bei dieser Ausgangssituation ist der SHELL SORT dem QUICK SORT überlegen!
Zeile 910: Nimmt das Suchwort entgegen
Zeile 920 bis 970: "High Low" Routine. Ganz wesentlich ist die Tatsache, daß wir die Suchwörter als "Wild Cards" definieren. Um in einer Kartei den Namen "ATARI" aufzufinden, genügt es, nur ein "A" als Suchwort einzutippen. Die Suchroutine bringt dann den ersten Datensatz, dessen Schlüsselwort mit "A" beginnt, auf den Bildschirm. Ist das nicht "ATARI", kann über die Funktion "VORWÄRTSBLÄTTERN" zu Atari gestoßen werden.
Zeile 1130 bis 1190: Vorwärtsblättern. Durch Betätigen Der Taste "+" wird der Datensatz mit der nächst größeren Indexnummer auf den Bildschirm gebracht.
Zeile 1200 bis 1250: Rückwärtsblättern. Durch Drücken der Taste "-" wird das Karteiblatt mit der nächst kleineren Indexnummer angezeigt.
Zeile 1270 bis 1390: Adressen löschen. Ist ein Datensatz gefunden, so kann über die Taste "L" die Adresse gelöscht werden.
Zeile 1330 bis 1390: Das Löschen einer Karteikarte geschieht einfach dadurch, daß ein Datensatz durch seinen Nachfolger überschrieben wird. Alle weiteren Nachfolger, sofern welche vorhanden sind, rücken um einen Index nach. Dadurch bleibt die ursprüngliche Sortierordnung erhalten, und der Schlüssel muß nicht gelöscht, und die Datei später nicht neu sortiert werden.

ADRESSEN LADEN/SPEICHERN
Eine bestehende Adressenkartei kann von Kassette oder Diskette geschrieben werden.
Zeile 1410 bis 1500: Adressen laden.
Zeile 1440: Zunächst wird nach dem Dateinamen gefragt.
Zeile 1470: Der Satz 0 der Datei ist mit der Karteiblätterzahl und dem Schlüssel, nach dem die Datei sortiert ist, belegt.
Zeile 1482: Jeder RECORD auf einem externen Gerät entspricht genau einem Adressenkarteiblatt.
Zeile 1520 bis 1610: Adressen speichern. Im Prinzip sind diese Zeilen identisch mit 1410 bis 1500 mit dem einzigen Unterschied, daß die Karteiblätter gespeichert werden.

ADRESSEN AUSDRUCKEN

Die gesamte Datei kann über einen Line-Printer ausgegeben werden.
Zeile 1670 bis 1800: Die Satznummer, mit welcher der Ausdruck beginnt und die, mit welcher er endet, können eingegeben werden.

UNTERPROGRAMME FÜR ADRESSEN

Zeile 1880 bis 2000: EINGABEMASKE. Dieses Programmsegment baut eine variable Bildschirmmaske für Ein- und Ausgabe von Adressen auf.
Zeile 2020 bis 2070: INPUT PROMPT. Nimmt ein ASCII-Zeichen von der Tastatur entgegen und überprüft, ob der Wert im Optionenstring enthalten ist. Die Position des Charakters im Optionenstring wird in T zurückgeliefert, wenn T 0 ist.
Zeile 2090 bis 2300: SHELL-SORT Sortiert die ersten N Elemente eines Feldes S$() nach dem Schlüssel S.
Zeile 2000: In der ersten praktischen Anwendung sehen wir, daß neben dem Schlüsselelement eine ganze Reihe weiterer Elemente (Zeile 2260) bewegt werden müssen. Eine gute und schnelle Sortierroutine zeichnet sich dadurch aus, daß für eine Sortierung eines Feldes so wenig Umstellungen als möglich vorgenommen werden müssen.

J. Peschetz

```
10 REM * DATENBANK FUER ADRESSEN*
20 REM * FUER CHIP-SPECIAL*
30 REM * VON J.PESCHETEZ *
100 DATA ***************************
*
110 DATA ***************************
*
120 DATA **{24SPACES}**
130 DATA **{4SPACES}D A T E N B A N
K{3SPACES}**
140 DATA **{24SPACES}**
150 DATA **{6SPACES}fuer ADRESSEN{5S
PACES}**
160 DATA **{24SPACES}**
170 DATA **========================*
*
180 DATA **{24SPACES}**
190 DATA ** 19840430 - J.PESCHETZ{2S
PACES}**
200 DATA **{24SPACES}**
202 DATA ***************************
*
204 DATA ***************************
*
210 DATA ,
220 DATA Adressen eingeben{7SPACES}(
1)
230 DATA Adressen suchen{9SPACES}(2)
240 DATA Adressen laden{10SPACES}(3)
250 DATA Adressen speichern{6SPACES}
(4)
260 DATA Adressen ausdrucken{5SPACES
}(5)
270 DATA Arbeit beenden{10SPACES}(6)
,
280 DATA EOF
290 REM
291 GRAPHICS 0
292 SAN=6:DIM SL(SAN)
293 SL(0)=0:SL(1)=SL(0)+25:SL(2)=SL(
1)+25:SL(3)=SL(2)+25:SL(4)=SL(3)+6:S
L(5)=SL(4)+25:SL(6)=SL(5)+10
294 SAL=SL(6)
300 NMAX=50:NM=NMAX*SAL
310 DIM S$(NM),M$(1),X$(SAL),OPT$(10
),OPT1$(10),OPT2$(10),OPT3$(10),OPT4
$(10)
320 SETCOLOR 2,0,0:OPEN #7,4,0,"K:"
375 S$(1)=" ":S$(NM)=S$(1):S$(2)=S$(
NM)
380 FOR J=1 TO SAL:S$(J)="?":NEXT J
390 PRINT CHR$(125):I=1:RESTORE 100:
POKE 752,0
400 READ X$:IF X$<>"EOF" THEN PRINT
X$:GOTO 400
405 PRINT "Ihre Wahl bitte{10SPACES}
";
410 OPT$="123456CHV":GOSUB 2050
411 IF M$<>"C" THEN 414
412 COL=COL+1:IF COL>15 THEN COL=0
413 SETCOLOR 2,COL,HEL:SETCOLOR 4,CO
L,HEL:GOTO 410
414 IF M$<>"H" THEN 417
415 HEL=HEL+1:IF HEL>15 THEN HEL=0
416 SETCOLOR 2,COL,HEL:SETCOLOR 4,CO
L,HEL:GOTO 410
417 IF M$<>"V" THEN 420
418 VOR=VOR+1:IF VOR>15 THEN VOR=0
419 SETCOLOR 1,O,VOR:GOTO 410
420 POKE 752,1:ON T GOSUB 550,670,14
40,1440,1670,440
430 GOTO 390
440 POKE 752,0:CLOSE #7:END
550 X$="{CODE193/196/210/197/211/211
/197/206/160/197/201/206/199/197/194
/197/206}":OPT1$="K(orrektur)?":OPT2
$="W(eiter)?":OPT3$="F(ertig)?":OPT4
$="":GOSUB 1900
560 I=N:S=0:NN=0
570 I=I+1:I1=I*SAL+1:POSITION 36,0:P
RINT I:SS=12
580 J=0:FOR ZZ=6 TO 16 STEP 2
585 J=J+1:LL=SL(J)-SL(J-1):GOSUB 254
00:S$(I1+SL(J-1),I1+SL(J))=X$
587 NEXT ZZ
590 OPT$="KWF":GOSUB 2050
600 ON T GOTO 580,570,610
610 N=I:RETURN
670 X$="{CODE193/196/210/197/211/211
/197/206/160/211/213/195/200/197/206
}":OPT1$="S(atz)?":OPT2$="K(riterium
)?":OPT3$="W(eiter)?":OPT4$="F(ertig
)?":GOSUB 1900
680 OPT$="SKWF+-L":GOSUB 2050
690 ON T GOTO 800,870,750,700,1160,1
230,1300
700 RETURN
750 T=T1:GOTO 690
800 T1=T
810 ZZ=0:SS=36:LL=3:NN=1:GOSUB 25400
:IF X<1 OR X>N THEN GOSUB 25760:GOTO
 810
820 I=X:GOSUB 2340:GOTO 680
870 T1=T
880 POSITION 2,19:PRINT "Schluessel?
{2SPACES}";
885 OPT$="123456":GOSUB 2050:PRINT T
;
890 IF T<>S THEN POSITION 20,19:PRIN
T "Sortieren";:S=T
895 S1=SL(S-1)+1:S2=SL(S):GOSUB 2120
900 POSITION 2,19:PRINT "{35SPACES}"
;
910 ZZ=19:SS=12:NN=0:LL=S2-S1+1
```

```
915 GOSUB 254ØØ:IF X$(1,1)=" " THEN
GOSUB 2576Ø:GOTO 91Ø
916 FOR LM=1 TO LEN(X$)
917 IF X$(LM,LM)=" " THEN POP :GOTO
92Ø
918 NEXT LM
92Ø LM=LM-1:K=Ø
93Ø I=N
94Ø J=INT((I+K)/2)
95Ø IF X$(1,LM)>S$(J*SAL+S1,J*SAL+S1
+LM-1) THEN K=J:GOTO 96Ø
955 I=J
96Ø IF K+1<I THEN 94Ø
97Ø IF X$(1,LM)<>S$(I*SAL+S1,I*SAL+S
1+LM-1) THEN I=Ø
98Ø GOSUB 234Ø:POSITION 12,19:PRINT
"{25SPACES}";
99Ø GOTO 68Ø
116Ø I=I+1:IF I>N THEN I=1
117Ø GOSUB 234Ø:GOTO 68Ø
123Ø I=I-1:IF I=Ø THEN I=N
124Ø GOTO 117Ø
13ØØ IF I=Ø THEN 68Ø
131Ø SOUND Ø,126,1Ø,8:SOUND 1,63,1Ø,
15
132Ø IF I=N THEN 137Ø
135Ø I1=I*SAL+1:I2=N*SAL:S$(I1,I2)=S
$(I1+SAL,I2+SAL)
137Ø N=N-1:SOUND 1,81,1Ø,15
138Ø I=Ø:GOSUB 234Ø
139Ø SOUND 1,96,1Ø,15:SOUND Ø,Ø,Ø,Ø:
SOUND 1,Ø,Ø,Ø:GOTO 68Ø
144Ø POSITION 25,15+T:PRINT ">":POSI
TION 2,23:PRINT "Dateiname?{6SPACES}
";
145Ø NN=Ø:ZZ=23:SS=14:LL=14:GOSUB 25
4ØØ
1455 IF T=4 THEN 157Ø
146Ø OPEN #1,4,Ø,X$
147Ø INPUT #1;N,S
148Ø FOR I=1 TO N
1482 INPUT #1;X$:S$(I*SAL+1)=X$
1486 NEXT I
149Ø CLOSE #1:RETURN
151Ø REM
157Ø OPEN #1,8,Ø,X$
158Ø PRINT #1;N;",";S
159Ø FOR I=1 TO N
16ØØ PRINT #1;S$(I*SAL+1,I*SAL+SAL)
161Ø NEXT I
162Ø GOTO 149Ø
167Ø POSITION 25,2Ø:PRINT ">";:POSIT
ION 2,23:PRINT "Start{2SPACES}1{1ØSP
ACES}";
1672 ZZ=23:SS=9:NN=1:LL=3:GOSUB 254Ø
Ø:IF X<1 THEN GOSUB 2576Ø:GOTO 1672
1674 I1=X
1676 POSITION 2Ø,23:PRINT "Ende{2SPA
CES}";N;
1678 SS=26:LL=3:GOSUB 254ØØ:IF X>N T
HEN GOSUB 2567Ø:GOTO 1678
169Ø OPEN #1,8,Ø,"P:"
17ØØ X$="{1ØSPACES}":PRINT #1;X$;"AD
RESSEN DATEI AUSDRUCKEN:":PRINT #1;X
$;"========================"
17Ø5 PRINT #1:PRINT #1
171Ø FOR I=I1 TO X
1715 I1=I*SAL
172Ø PRINT #1;X$;"Adresse ";STR$(I);
":":PRINT #1;X$;
1725 FOR J=1 TO LEN(STR$(I))+9:PRINT
 #1;"-";:NEXT J:PRINT #1:PRINT #1
173Ø PRINT #1;X$;"Name:",S$(I1+1,I1+
SL(1))
174Ø PRINT #1;X$;"Vorname?",S$(I1+SL
(1)+1,I1+SL(2))
175Ø PRINT #1;X$;"Strasse:",S$(I1+SL
(2)+1,I1+SL(3))
176Ø PRINT #1;X$;"Plz.:",S$(I1+SL(3)
+1,I1+SL(4))
177Ø PRINT #1;X$;"Stadt:",S$(I1+SL(4
)+1,I1+SL(5))
178Ø PRINT #1;X$;"Tel.Nr.:",S$(I1+SL
(5)+1,I1+SL(6))
179Ø PRINT #1:PRINT #1
18ØØ NEXT I:CLOSE #1:RETURN
19ØØ PRINT CHR$(125):POSITION 2,Ø:PR
INT X$:POSITION 31,Ø:PRINT "SATZ":PO
SITION 2,1
19Ø5 FOR II=1 TO 36:PRINT CHR$(21);:
NEXT II
191Ø POSITION 2,6:PRINT "{CODE2Ø6/22
5/237/229/186}"
192Ø POSITION 2,8:PRINT "{CODE214/23
9/242/238/225/237/229/186}"
193Ø POSITION 2,1Ø:PRINT "{CODE211/2
44/242/225/243/243/229/186}"
194Ø POSITION 2,12:PRINT "{CODE2Ø8/2
36/25Ø/174/186}"
195Ø POSITION 2,14:PRINT "{CODE211/2
44/225/228/244/186}"
196Ø POSITION 2,16:PRINT "{CODE212/2
29/236/174/2Ø6/242/174/186}"
197Ø POSITION 2,2Ø:FOR II=1 TO 36:PR
INT CHR$(18);:NEXT II
198Ø POSITION 2,21:PRINT OPT1$:POSIT
ION 28,21:PRINT OPT2$
199Ø POSITION 2,22:PRINT OPT3$:POSIT
ION 28,22:PRINT OPT4$
2ØØØ RETURN
2Ø5Ø GET #7,T:M$=CHR$(T)
2Ø6Ø FOR T=1 TO LEN(OPT$)
2Ø62 IF OPT$(T,T)=M$ THEN TON=57:GOS
UB 254Ø:RETURN
```

Adress10.BAS
Adressen.DAT

```
2Ø64 NEXT T
2Ø66 GOSUB 2576Ø:GOTO 2Ø5Ø
212Ø IF N>2 THEN 216Ø
2125 IF N<2 THEN RETURN
213Ø IF S$(SAL+S1,SAL+S2)<=S$(2*SAL+
S1,2*SAL+S2) THEN RETURN
214Ø X$=S$(SAL+1,2*SAL):S$(SAL+1)=S$
(2*SAL+1,3*SAL):S$(2*SAL+1)=X$
215Ø RETURN
216Ø R=N-1:N1=N*SAL+1:M1=SAL-1:SA=S1
-1:SE=S2-1
217Ø R=INT(R*Ø.3+Ø.5):R1=R*SAL
218Ø FOR K=1 TO R1+1 STEP SAL
219Ø FOR I=K+R1 TO N1 STEP R1
22ØØ X$=S$(I,I+M1)
221Ø FOR J=I-R1 TO 1 STEP -R1
223Ø IF X$(S1,S2)>S$(J+SA,J+SE) THEN
 POP :GOTO 226Ø
224Ø S$(J+R1,J+R1+M1)=S$(J,J+M1)
225Ø NEXT J
226Ø S$(J+R1,J+R1+M1)=X$
227Ø NEXT I:NEXT K
228Ø IF R>1 THEN 217Ø
229Ø RETURN
234Ø POSITION 36,Ø:PRINT "{3SPACES}"
;
235Ø POSITION 36,Ø:PRINT I;:I1=I*SAL
:J=Ø:SS=12
236Ø FOR ZZ=6 TO 16 STEP 2
2365 J=J+1:POSITION SS,ZZ:PRINT S$(I
1+SL(J-1)+1,I1+SL(J));
2367 NEXT ZZ
237Ø RETURN
254Ø SOUND Ø,TON,1Ø,13
255Ø FOR II=1 TO 1Ø:NEXT II
256Ø SOUND Ø,Ø,Ø,Ø:RETURN
254ØØ TON=72:GOSUB 254Ø:IF LL=Ø THEN
 RETURN
2541Ø NLL=ABS(LL):X$=" ":X$(NLL)=X$(
1):X$(2)=X$(1)
2542Ø POSITION SS,ZZ
2543Ø IF LL<Ø THEN PRINT X$;
2544Ø SE=SS+NLL-1:IF SE>39 THEN SE=3
9
25445 SI=SS
2545Ø LOCATE SI,ZZ,CHR1:POSITION SI,
ZZ:PRINT CHR$(95);
25455 POSITION SI,ZZ:IF SI=SE THEN P
RINT CHR$(CHR1);:GOTO 2546Ø
25456 PRINT CHR$(95);
2546Ø GET #7,CHAR:POSITION SI,ZZ:PRI
NT CHR$(CHR1);
2547Ø IF CHAR=155 THEN 257ØØ
2548Ø IF CHAR<>3Ø THEN 2551Ø
2549Ø SI=SI-1:IF SI<SS THEN SI=SS
255ØØ GOTO 2545Ø
2551Ø IF CHAR<>31 THEN 2554Ø
2552Ø SI=SI+1:IF SI>SE THEN SI=SE
2553Ø GOTO 2545Ø
2554Ø IF CHAR<>255 THEN 2557Ø
2557Ø IF CHAR<>254 THEN 2559Ø
2559Ø POSITION SI,ZZ:PRINT CHR$(CHAR
);
256ØØ SI=SI+1:IF SI>SE THEN SI=SE
2561Ø GOTO 2545Ø
2567Ø REM
257ØØ X$="":LL=Ø:FOR SI=SS TO SE:LL=
LL+1
2571Ø LOCATE SI,ZZ,CHAR:IF CHAR=95 T
HEN POP :GOTO 25725
25715 X$(LL,LL)=CHR$(CHAR)
2572Ø NEXT SI
25725 TRAP 2575Ø
2573Ø IF NN<>Ø THEN X=VAL(X$)
2574Ø TRAP 44444:RETURN
2575Ø TRAP 44444:GOSUB 2576Ø:GOTO 25
44Ø
2576Ø SOUND Ø,145,1Ø,15
2577Ø SOUND 1,14Ø,1Ø,13
2578Ø SOUND 2,217,5,8
2579Ø FOR TL=1 TO 5Ø:NEXT TL
258ØØ SOUND Ø,Ø,Ø,Ø:SOUND 1,Ø,Ø,Ø:SO
UND 2,Ø,Ø,Ø
2581Ø RETURN

  READY !
```

Laterna Magica

AI Nr. 0

Beschreibung des Programmes (Atari 600XL/800XL, 1220 Bytes)
Dieses Programm zaubert eine entzückende Version des sogenannten Kaleidoskops auf den Bildschirm des ATARI-Computers.
Niemand kann die überraschenden und faszinierenden Farbkombinationen voraussagen. Alles funktioniert wie bei dem von der Kindheit bekannten Vorbild. Ein Kindertraum wird wahr.

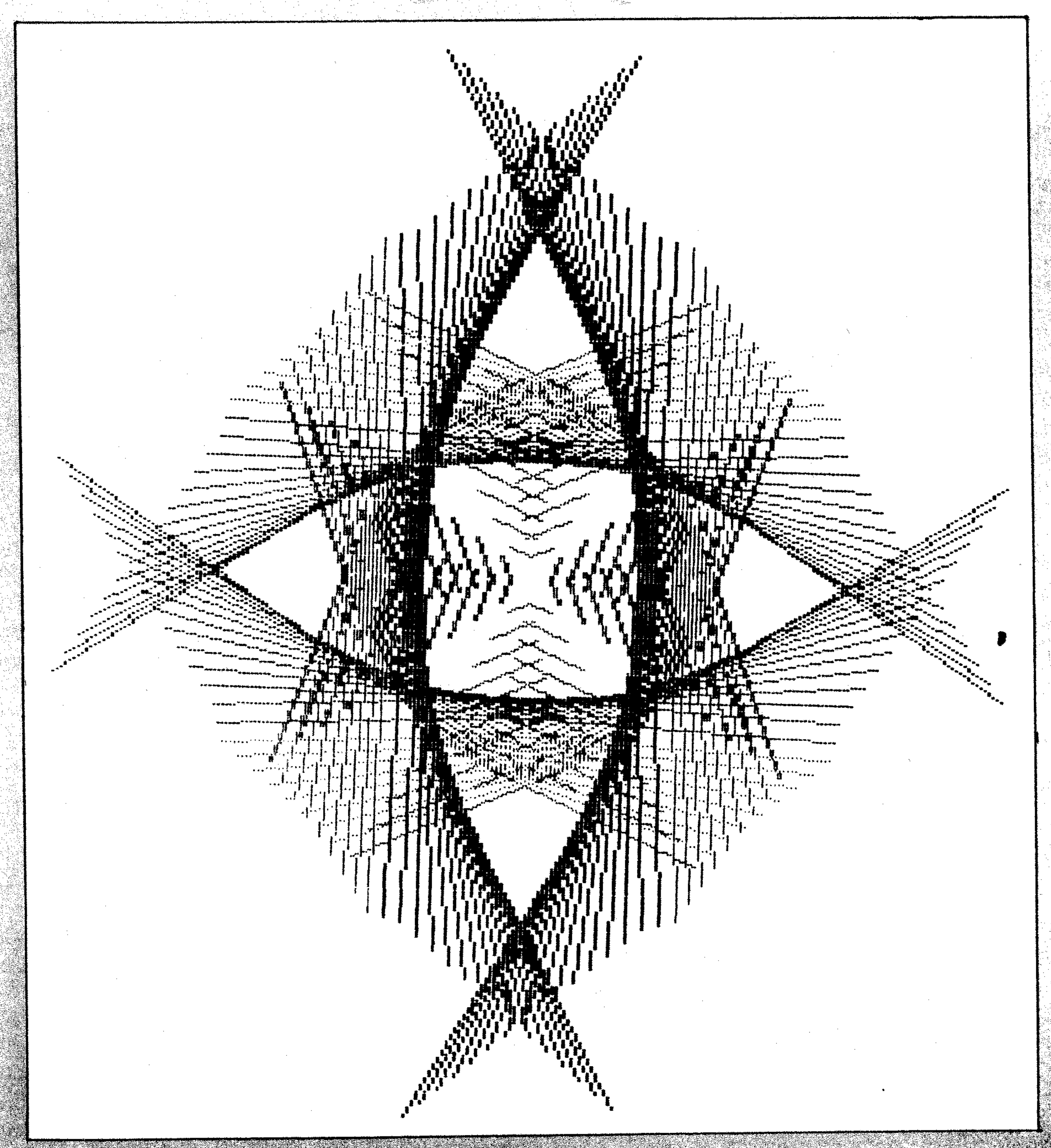
Benutzt wird der Graphics-Mode 15+16 des ATARI-Computers. Das bedeutet, daß der gesamte Bildschirm (ohne Textfenster) in 192 Zeilen

und 160 Spalten aufgelöst wird und in vier verschiedenen Farben gleichzeitig angesprochen werden kann. Das ist zwar nicht hochauflösend, aber gut genug für unsere Zwecke. Und dieses kleine Programm demonstriert auf ganz überzeugende Art und Weise die überragenden grafischen Fähigkeiten des ATARI-Computers.

PROGRAMM-AUFBAU

In den Zeilen 250 bis 260 werden die Startbedingungen aufgestellt, (WH,MI) Zentrum des Koordinatensystems. Zeile 267 bis 280: Eine Linie und ihr Spiegelbild ziehen. Zeile 270: Diese Fehlerfalle erspart uns das Testen der Randbezirke. Liegt ein Punkt außerhalb des Bildschirmes, so verursacht das Auftreten eines Bereichfehlers einfach die Rückkehr vom Unterprogramm. Zeile 290 bis 300: Neue Start- und Endpunkte von Linien berechnen. Zeile 310 bis 320: Linien zeichnen und um X- und y-Achse spiegeln. Zeile 330: Bedingung für "Linien brechen". Zeile 340: Bedingung für Start eines neuen Musters. Zeile 356 bis 380: Linien "brechen" und neue Farbe festlegen. Zeile 386 bis 420: Bildschirm löschen und Start eines neuen Musters.

J. Peschetz



MONITORAUFNAHME KALEIDOSKOP

```
100 REM *LATERNA MAGICA*
110 REM *FUER CHIP-SPECIAL*
120 REM *VON J.PESCHETZ*
250 GRAPHICS 15+16
255 MI=80:M1=MI+1:WH=MI:A=2
260 COL=INT(7*RND(1)+1):GOTO 390
270 TRAP 282:COLOR COL:PLOT X1+WH,Y1
+MI:DRAWTO X2+WH,Y2+MI
280 PLOT Y1+WH,X1+MI:DRAWTO Y2+WH,X2
+MI
282 RETURN
290 H=X1+XV1:X1=H-INT(H/M1)*M1:H=Y1+
YV1:Y1=H-INT(H/M1)*M1
300 H=X2+XV2:X2=H-INT(H/M1)*M1:H=Y2+
YV2:Y2=H-INT(H/M1)*M1
310 Y1=-Y1:Y2=-Y2:GOSUB 270:X1=-X1:X
2=-X2:GOSUB 270
320 Y1=-Y1:Y2=-Y2:GOSUB 270:X1=-X1:X
2=-X2:GOSUB 270
330 IF RND(0)*A<1 THEN GOSUB 360
340 IF RND(0)*150<1 THEN 390
350 GOTO 290
360 XV1=INT(RND(0)*11+0.5)-5:YV1=INT
(RND(0)*11+0.5)-5
370 XV2=INT(RND(0)*11+0.5)-5:YV2=INT
(RND(0)*11+0.5)-5
380 COL=INT(RND(0)*7+1):A=10
382 RETURN
390 GRAPHICS 15+16:A=1
400 X1=1:Y1=1
410 X2=1:Y2=1
420 GOTO 310

READY !
```

List-Format

Beschreibung des Programmes (1994 Bytes, Atari 600 XL/800 XL, Kassetten- oder Diskettenlaufwerk und Drucker)

Gibt man normalerweise ein Programmlisting über den an den Atari-Computer angeschlossenen Drucker aus, so geht das zwar problemlos über LIST "P: oder LIST "R: Aber die Sache hat dennoch einen Haken: Die auf dem Drucker ausgegebene Anzahl der Zeichen/Druckseite (meist 72) stimmt nicht mit der Anzahl der Zeichen/Bildschirmzeile überein. Dies erschwert das Eintippen des Programmmes in den Computer. Zudem ist das Seitenformat des Listings nicht besonders ansprechend, da der Drucker über die Seitenperforation hinausdruckt.

Unser Programm LIST-FORMAT umschifft diese Klippen und stellt noch einige zusätzliche Funktionen bereit:

* Herstellen eines beliebig formatierten Programmlistings
* Löschen einer größeren Anzahl von Zeilen aus einem BASIC-Programm
* Herauskopieren von Zeilen aus einem BASIC-Programm.

Diese Funktionen sind sehr hilfreich, da der im Atari-Computer eingebaute Editor nicht gerade einer der komfortabelsten ist.

ARBEITSWEISE DES PROGRAMMES

Nehmen wir an, wir möchten ein Programm namens "BAUERN.BAS" mit dem Programm "LFORMAT.BAS" formatieren. Folgende Schritte sind notwendig, nachdem Diskettenlaufwerk und Drucker betriebsbereit sind:

Abspeichern des Programmes als File LIST "BAUERN.LST" auf Diskette (oder Kassette LIST "C"). Anstelle des oben angeführten Dateinamens kann ein beliebiger gültiger Dateinamen treten. Laden und Starten des Formatierprogrammes mit RUN"LFORMAT.BAS" von Diskette oder Kassette (RUN "C:"). Auf dem Bildschirm erscheint nach kurzer Zeit die Bildschirmmaske, deren Fragen wir auch gleich für unser Beispiel beantwortet haben.

Name des Programmes: Jeder gültige Dateiname, der eine ASCII-Datei enthält

Ausgabedateiname: Ein Gerätename oder ein gültiger Dateiname (es kann über Drucker, Bildschirm, Diskette oder Kassette ausgegeben werden).

Zeilen-Nr.: Wird nur die Returntaste betätigt, werden alle Zeilen des Programmes formatiert, sonst von der Start- zur Endzeile.

Zeichen/Zeile: Zeilen, die länger

als die hier angegebene Buchstabenzahl sind, werden in so viele Zeilen aufgebrochen wie erforderlich.
Zeilen/Seite: Nach der hier eingegebenen Zeilenanzahl wird ein Formfeed gesendet. Bei der Formatierung wird davon ausgegangen, daß ein Papier im DIN-A4-Format (72 Zeilen-/Seite) eingepannt ist.

PROGRAMMAUFBAU

Zeile 100 bis 320: Aufbau der Bildschirmmaske und Entgegennahme der Parameter.
Zeile 265: Das Programm bleibt hier solange "hängen", bis ein gültiger Geräte-/ Dateiname der Variablen FILE1$ zugeordnet ist (Eingabedatei).
Zeile 285: Wie vorher, nur mit Bezug auf FILE2$ und Ausgabedateinamen.
Zeile 292: Zeilenparameter entgegennehmen. Die Betätigung der Returntaste ohne Eingabe von Parametern führt zu einem Fehler und zur Annahme, daß alle (!) Programmzeilen formatiert werden sollen.
Zeile 300 bis 312: Eingabe der Zeichen/Zeile.
Zeile 314 bis 320: Eingabe der Zeilen/Seite.
Zeile 324 bis 340: Formatieren
Zeile 345: Tritt ein Lesefehler (Dateiende!) auf, ist die Arbeit beendet.
Zeile 352 bis 353: Ermittelt die Zeilennummern der BASIC-Programmzeilen.
Zeile 356 bis 357: Formatierungsanfang und -ende testen.
Zeile 360: Hier ist die Programmzeile kürzer als das Format Zeichen/Zeile.
Zeile 370 bis 420: Aufbrechen einer längeren Programmzeile auf das gewünschte Format; L = Zeichen/Zeile.
Zeile 510 bis 540: Programmabbruch - Bedingung checken.
Zeile 970 bis 1040: Erzwingt einen Seitenvorschub, wenn mehr Zeilen vorliegen als im Format begrenzt wurden.

Wird das Programm zum Herausheben oder Löschen eines Programmzeilenblockes verwendet, so muß der Sprung zum Seitenvorschub (GOSUB 1000) entfallen und als Ausgabedatei eine Disketten- oder Kassettenkartei gewählt werden. Eine BASIC-Programmzeile kann 138 Zeichen aufnehmen. Das Rumpfprogramm liegt dann als Datenfile vor und kann mit "ENTER" geladen werden.

J. Peschetz

```
10 REM * LISTFORMAT*
20 REM * FUER CHIP-SPECIAL*
30 REM * VON J.PESCHETZ*
100 DATA ***************************
110 DATA ***************************
120 DATA **{23SPACES}**
130 DATA **{2SPACES}L I S T F O R M
A T{2SPACES}**
140 DATA **{23SPACES}**
150 DATA **=======================**
160 DATA **{23SPACES}**
170 DATA ** J. PESCHETZ{2SPACES}1984
0408 **
180 DATA **{23SPACES}**
190 DATA ***************************
195 DATA ***************************
200 DATA ,
210 DATA EOF
220 DIM M$(114),FILE1$(14),FILE2$(14
)
225 SETCOLOR 2,0,0
230 RESTORE :PRINT CHR$(125)
240 READ M$:IF M$<>"EOF" THEN PRINT
M$:GOTO 240
250 PRINT "NAME DES PROGRAMMES{2SPAC
ES}";
260 INPUT FILE1$
265 TRAP 267:OPEN #1,4,0,FILE1$:GOTO
 270
267 CLOSE #1:GOTO 250
270 PRINT "AUSGABEDATEINAME{5SPACES}
";
280 INPUT FILE2$
285 TRAP 287:OPEN #2,8,0,FILE2$:GOTO
 290
287 CLOSE #2:GOTO 270
290 IF FILE1$=FILE2$ THEN 250
292 TRAP 293:PRINT "ZEILEN Nr. (von,
bis) ";:INPUT ZA,ZE:GOTO 294
293 ZA=0:ZE=44444
294 TRAP 44444:ZA=ABS(INT(ZA)):ZE=AB
S(INT(ZE))
```

```
295 IF ZE<ZA THEN 292
300 PRINT "ZEICHEN/ZEILE (38){3SPACE
S}";
310 INPUT L
312 IF L<10 OR L>114 OR L-INT(L)<>0
THEN 300
314 PRINT "ZEILEN/SEITE{2SPACES}(45)
{3SPACES}";
316 INPUT ZS
320 IF ZS<10 OR ZS>72 OR ZS-INT(ZS)<
>0 THEN 314
342 S=0
345 TRAP 500
350 INPUT #1,M$:LM=LEN(M$)
352 NRL=7
353 TRAP 353:NRL=NRL-1
355 NR=VAL(M$(1,NRL))
356 IF NR<ZA THEN 345
357 IF NR>ZE THEN GOTO 500
360 IF LM<=L THEN PRINT #2,M$:S=S+1:
GOSUB 1000:GOTO 345
370 I=1
380 J=I+L-1
390 IF J>LM THEN J=LM
400 PRINT #2,M$(I,J)
405 S=S+1:GOSUB 1000
410 I=I+L:IF I<=LM THEN 380
420 GOTO 345
500 TRAP 44444:CLOSE #1:CLOSE #2
510 PRINT :PRINT "Weiter formatieren
 (J/N) ";
520 INPUT M$
530 IF M$="J" OR M$="j" THEN 230
540 STOP
1000 IF S<ZS THEN RETURN
1010 FOR Z=1 TO 72-ZS
1020 PRINT #2
1030 NEXT Z
1040 S=O:RETURN
  READY !
```

Fehler-Melder

Beschreibung des Programmes (Atari 600XL/800XL, 4770 Bytes)
Das Error-Melder-Programm stellt eine Erleichterung für Anfänger dar, die sich im Atari-Basic noch nicht auskennen und mit den Fehlermeldungen nichts anfangen können. Jede Fehlermeldung während des Programmablaufes wird nun durch das Fehler-Hilfs-Programm gleich in eine Erklärung umgewandelt. Am Anfang des Programmes sollte die Anweisung TRAP 30000 stehen. Nun kann der Anwender anfangen zu programmieren. Dieses Programm sollte auf Kassette oder Diskette abgespeichert werden und bevor man anfängt, ein Programm zu schreiben, eingeladen werden. Als erste Zeile geben Sie ein: 0 TRAP 30000.

Wolfgang Schmidt

```
0 TRAP 30000
29999 END
30000 GRAPHICS 0:? :? :? "***{2SPACE
S}FEHLER#";PEEK(195);" eingegangen *
**{CODE253}":ERN=PEEK(195):RESTORE 3
1000+ERN:PRINT
30001 ERRLINE=PEEK(187)*256+PEEK(186
):TRAP 30010:DIM ERROR$(80)
30002 REM ***************************
*
30003 REM *{25SPACES}*
30004 REM *FEHLER-MELDUNGEN(DEUTSCH)
*
30005 REM *{25SPACES}*
30006 REM * FUER CHIP-SPECIAL{7SPACE
S}*
30007 REM * VON MICHEAL SCHMIDT{5SPA
CES}*
30008 REM *{25SPACES}*
30009 REM ***************************
*
30010 READ ERROR$:IF PEEK(183)+256*P
EEK(184)-31000<>ERN THEN PRINT "< UN
DEFINIERTER FEHLER AUFGETRETEN >":EN
D
30015 PRINT ERROR$
30016 PRINT :PRINT "Folgende Zeile h
at die Fehler-Meldung verursacht:"
30017 LIST ERRLINE:REM * 16.01.1984
*
30020 REM ***************************
*
```

```
CES}*
30022 REM * deines Programms{8SPACES
}*
30023 REM * TRAP 30000 und jede{5SPA
CES}*
30024 REM * Fehlermeldung in deinem
*
30025 REM * Programm wird in deutsch
*
30026 REM * ausgegeben, die fehler-
*
30027 REM * hafte Zeile wird gezeigt
*
30028 REM * Deine verwendeten Zeilen
* 30029 REM * nrn muessen < 30000 se
in*
30030 REM ***************************
*
31000 REM {CODE157} FEHLERMELDUNGEN
{CODE157
31002 DATA Zu wenig Speicherplatz
31003 DATA Wert liegt ausserhalb des
 zul. Ber.
31004 DATA Zu viele Variablen.(Max.1
28 erlaubt)
31005 DATA Stringlaengenfehler
31006 DATA Kein DATA-Element mehr fu
er READ.
31007 DATA Zeilennummer>32767 oder n
egativ.
31008 DATA Fehler im INPUT. Versuch
eine nicht- numerische Variable in e
ine num. Var. einzulesen.
31009 DATA Feld- oder String-DIM-Feh
ler.{9SPACES}Unzulaessigen Wert verw
endet.
31010 DATA Zu viele GOSUBS oder Ausd
ruecke
31011 DATA Gleitkomma-Ueberlauf.{17S
PACES}Zahl> 1E+98 oder < 1E-99
31012 DATA Zeile nicht gefunden. GOT
O/ GOSUB oderTHEN zu einer nicht exi
st. Zeile.
31013 DATA Kein zugehoeriges 'FOR'.{
14SPACES}NEXT ohne FOR wurde gefunde
n.
31014 DATA Zeile zu lang. Zu lange o
der komplexe Anweisungen.
31015 DATA 'GOSUB' oder 'FOR' Zeile
wurde{8SPACES}geloescht.
31016 DATA RETURN-Fehler.{24SPACES}R
ETURN ohne vorheriges GOSUB gefunden
.
31017 DATA Syntax-Fehler. Unverstaen
dliche{7SPACES}Eingabe in einer Zeil
e.
31018 DATA Falsches Textzeichen.{17S
PACES}VAL-Befehl auf nicht-num. Wert
 angew.
31019 DATA LOAD Programm zu lang.{16
SPACES}Zu wenig freier RAM fuer ganz
es Progr.
31020 DATA E/A-Nummer falsch.{20SPAC
ES}Sie muss zwischen 1 und 7 liegen.
31021 DATA Datei-Ladefehler.{21SPACE
S}And. Format abgespeichert/nicht le
sbar
31128 DATA Abbruch durch BREAK waehr
end einer{4SPACES}E/A-Operation
31129 DATA IOCB schon offen.{21SPACE
S}Zweites OPEN des selben Files.
31130 DATA Nicht vorhandenes E/A-Ger
aet angespr. Name nicht vorhanden.
31131 DATA IOCB nur zum Schreiben -{
14SPACES}nicht zum Lesen geoeffnet.
31132 DATA Unzulaessiger Handler-Bef
ehl.{9SPACES}Falscher IOCB-Befehl.
31133 DATA E/A-Geraet / Datei nicht
offen.{7SPACES}OPEN fehlt - oder fal
sche #.
31134 DATA Falsche IOCB-Nummer.{18SP
ACES}Zahl muss zwischen 1 und 7 sein
.
31135 DATA IOCB nur zum Lesen - nich
t zum{8SPACES}Schreiben offen oder s
chreiben unmoegl
31136 DATA Ende der Datei.{23SPACES}
Keine Daten mehr zu lesen.
31137 DATA Datei abgeschnitten.{18SP
ACES}Datenformat > 119 Bytes.
31138 DATA E/A-Geraet Zeitabschaltun
g.{11SPACES}Geraet antwortet nicht.
31139 DATA E/A-Geraet negatives Quit
tunssignal.{2SPACES}Diverse Fehler.
31140 DATA Serieller Rahmenbit-Fehle
r.{11SPACES}Kommunikation mit Periph
eri gestoert.
31141 DATA Cursor ausserhalb des zul
. Bereichs.{2SPACES}Position pruefen
.
31142 DATA Serieller Overrun.{20SPAC
ES}Daten kommen zu schnell herein.
31143 DATA Pruefsummenfehler. Daten
verstuemmelt.Nochmal probieren.
31144 DATA E/A-Geraete-Ausfuehrungsf
ehler.{7SPACES}Geraet kann d. Befehl
 nicht ausfuehren/Disk schreibgeschu
etzt.
31145 DATA Unzulaessiger Graphik-Mod
us/{10SPACES}Fehler beim Test nach s
chreiben.
31146 DATA Funktion nicht vorgesehen
.{12SPACES}Handler kann den Befehl n
```

```
icht ausf.
31147 DATA Kein ausreichender RAM fu
er diese{5SPACES}Graphikstufe vorhan
den.
31150 DATA Laufwerknummer-Fehler.{16
SPACES}Falsche Nummer oder kein Ansc
hluss.
31161 DATA Zu viele offene Dateien.{
14SPACES}Max. 8 gleichzeitig offen.
31162 DATA Disk voll - Daten koennen
 nicht mehr{2SPACES}uebertragen werd
en.
31163 DATA Fataler E/A-Fehler.{19SPA
CES}Fehlerhaftes DOS oder Disk.
31164 DATA Dateinummer vertauscht.{1
5SPACES}Fehlerhafte Disksektor-Verke
ttung.
31165 DATA Dateinamen-Fehler.{20SPAC
ES}Name enthaelt illegale Zeichen.
31166 DATA Fehler beim POINT-Befehl.
{13SPACES}Bytezaehler > 125/253.
31167 DATA Datei(name) gesperrt. (Fi
le locked)
31168 DATA Ungueltiger E/A-Geraete-B
efehl.{7SPACES}Unzulaessigen Befehl
erteilt.
31169 DATA Inhaltsverzeichnis Disk (
Directory){3SPACES}ist voll. Anzahl
> 64.
31170 DATA Datei nicht gefunden.{17S
PACES}Datei nicht im Inhaltsverzeich
nis.
31171 DATA Ungueltiger POINT-Befehl.
{13SPACES}Falsche POINT-Parameter.
31172 DATA APPEND-Befehl hier nicht
zulaessig.{3SPACES}DOS 1 und DOS 2 g
emixt.
31173 DATA Schlechte Sektoren beim F
ormatieren{3SPACES}einer Disk entdec
kt.

 READY !
```

Verschlungene Pfade

Beschreibung des Programmes (Atari 600XL/800XL,ein Jockstick, 8465 Bytes)

Neben Schnelligkeit ein gehöriges Maß an Überblick erfordert dieses spannende Labyrinth-Spiel. Dazu gehört folgende Story:

Dynamit-Joe ist Aufseher in einem Silberbergwerk. Bei einem seiner Kontrollgänge entdeckt er, daß in einem der Vorratsbunker ein Feuer ausgebrochen ist. Vermutlich durch eine der achtlos weggeworfenen Zigarettenkippen. Jetzt besteht die Gefahr, daß sich die eingelagerten Dynamit-Stangen entzünden und der ganze Berg in die Luft fliegt. Unser Held kann die Mine nur retten, indem er die schweren, glimmenden Stangen zum nächsten Wasseranschluß trägt und dort löscht. Aber der Weg dorthin ist lang und beschwerlich. Und zu allem Überfluß behindern Dynamit-Joe die großen, lediglich zur Sicherheit installierten eisernen Drehtore.

SPIELANLEITUNG

Erforderlich ist ein Joystick in Port 1. Nach Eingabe des Programms wird mit RUN gestartet. Nachdem das Titelbild erschienen ist, kann mit dem Feuerknopf oder der START-Taste begonnen werden.

Dynamit-Joe startet am Wasserhahn. Sobald sich eine Dynamit-Stange am oberen linken Spielfeldrand entzündet hat, muß er vor die Stange laufen, sie aufnehmen (Joystick nach links) und so schnell es geht zum Wasserhahn laufen. Dort kann er die Stange löschen (Joystick nach rechts). Für die ganze Aktion hat er 100 Einheiten Zeit. Braucht er länger, dann explodiert das Dynamit. Die Tore können nur dann durchlaufen werden, wenn mindestens ein Flügel offen steht. Sind beide Flügel geschloßen, dann ist der Weg versperrt und Dynamit-Joe muß sich einen anderen Weg suchen. Ist ein Tor durchlaufen, dann drehen sich die oder der zurückliegende Flügel um 90 Grad. Das bedeutet, ein Tor kann für den Rückweg versperrt sein. In jeder Runde müssen zehn Stangen Dynamit gelöscht werden. Nach jedem Löschen taucht ein Torflügel zusätzlich an einer wichti-

gen Stelle auf, der Weg wird immer komplizierter. Nach vier Runden ist die Mine gerettet.

VARIABLENTABELLE

ZS = Zeitsumme
ZEIT = Computer-Zeit
ZA = angezeigte Zeit
NO = Zeiger ob Durchgang frei
POS = Joes Position
VOR = Position vorne
VR = Position vorne rechts
VL = Position vorne links
ZUST = Joes Zustand
DY = Dynamit-Stangen
SP = Bildschirmposition
PE = Punkte pro Dynamit-Stangen
RD = Runde
ST = Startadresse
GP = Gesamtpunktzahl
TOR = Tor
BO = Bonus
ML = Melodie bei Bonus
TN = Ton
LE = Länge des Tons
I,X = Zählvariablen
A = Readvariable
P,Q = Pausenvariablen
V1,V2,V3 = Lautstärken Sound 0-2

PROGRAMM-AUFBAU

40-100: Hauptschleife, 110-150: Figurenbewegung nach unten, 160-200: Figurenbewegung nach oben. 210-260: Figurenbewegung nach links. 270-290: Aufnahme der Dynamit-Stange. 300-350: Figurenbewegung nach rechts. 360-390: Abfrage, ob Weg frei. 400-420: Umklappen der durchschrittenen Tore. 430-540: Löschen des Dynamits und setzen eines neuen Tores. 550-580: Neue Runde, Verteiler. 590-650: Bonus-Routine. 660-690: Rundenanzeige. 700-710: Verteiler für neue Runde. 720-750: Setzen der Tore für erste Runde. 760-790: Setzen der Tore für zweite Runde. 800-830: Setzen der Tore für dritte Runde. 840-870: Setzen der Tore für die nun folgende vierte Runde. 880-1050: Titelbild/Aufbau und Bewegung. 1060-1140: Aufbau des allgemeinen Spielfeldes. 1150-1210: Maschinensprache-Unterprogramm. 1220-1340: Datas der neuen Zeichen. 1350-1500: Explosionsroutine, Anzeige Spielende.

Axel Belz

```
10 GOSUB 1150
20 GOTO 880
30 POKE 19,0:POKE 20,0:POKE 77,0:ZS=
0
39 REM {CODE200/193/213/208/212/211/
195/200/204/197/201/198/197}
40 ZEIT=PEEK(20)+256*PEEK(19):ZA=INT
(ZEIT/20):IF ZA>100 THEN 1350
50 SOUND 0,0,0,0:POSITION 5,11:? #6;
ZA
60 IF STICK(0)=13 THEN GOSUB 110
70 IF STICK(0)=14 THEN GOSUB 160
80 IF STICK(0)=11 THEN GOSUB 210
90 IF STICK(0)=7 THEN GOSUB 300
100 NO=0:GOTO 40
109 REM {CODE198/201/199/213/210/160
/210/213/206/212/197/210}
110 VOR=POS+20:VR=POS+19:VL=POS+21:G
OSUB 360:IF NO=1 THEN RETURN
120 POKE POS,0:POS=POS+20:POKE POS,Z
UST
130 I=-19:GOSUB 400:I=-21:GOSUB 400
140 SOUND 0,29,12,8
150 RETURN
159 REM {CODE198/201/199/213/210/160
/200/207/195/200}
160 VOR=POS-20:VR=POS-19:VL=POS-21:G
OSUB 360:IF NO=1 THEN RETURN
170 POKE POS,0:POS=POS-20:POKE POS,Z
UST
180 I=19:GOSUB 400:I=21:GOSUB 400
190 SOUND 0,29,12,8
200 RETURN
209 REM {CODE198/201/199/213/210/160
/204/201/206/203/211}
210 IF PEEK(POS-1)=198 THEN GOSUB 27
0:RETURN
220 VOR=POS-1:VR=POS-21:VL=POS+19:GO
SUB 380:IF NO=1 THEN RETURN
230 POKE POS,0:POS=POS-1:POKE POS,ZU
ST
240 I=-19:GOSUB 400:I=21:GOSUB 400
250 SOUND 0,29,12,8
260 RETURN
269 REM {CODE193/213/198/206/193/200
/205/197}
270 ZUST=193:POKE POS,ZUST:SOUND 1,2
0,12,10:FOR P=1 TO 10:NEXT P:SOUND 1
,0,0,0
280 DY=DY+1:POSITION 0,1:? #6;10-DY:
IF DY=10 THEN POKE SP+21,0:POKE SP+2
2,67:RETURN
290 POKE SP+21,4:POKE SP+22,5:RETURN
299 REM {CODE198/201/199/213/210/160
```

```
/21Ø/197/195/2ØØ/212/211}
3ØØ IF PEEK(POS+1)=138 AND ZUST=193
THEN 43Ø
31Ø VOR=POS+1:VR=POS+21:VL=POS-19:GO
SUB 38Ø:IF NO=1 THEN RETURN
32Ø POKE POS,Ø:POS=POS+1:POKE POS,ZU
ST
33Ø I=-21:GOSUB 4ØØ:I=19:GOSUB 4ØØ
34Ø SOUND Ø,29,12,8
35Ø RETURN
359 REM {CODE215/197/199/16Ø/198/21Ø
/197/201/16Ø/191}
36Ø IF PEEK(VOR)<>Ø OR PEEK(VR)=66 A
ND PEEK(VL)=66 THEN NO=1
37Ø RETURN
38Ø IF PEEK(VOR)<>Ø OR PEEK(VR)=67 A
ND PEEK(VL)=67 THEN NO=1
39Ø RETURN
399 REM {CODE212/2Ø7/21Ø/197/16Ø/213
/2Ø5/2Ø3/2Ø4/193/2Ø8/2Ø8/197/2Ø6}
4ØØ IF PEEK(POS+I)=67 THEN POKE POS+
I,66:RETURN
41Ø IF PEEK(POS+I)=66 THEN POKE POS+
I,67
42Ø RETURN
43Ø ZUST=129:POKE POS,ZUST
439 REM {CODE2Ø4/2Ø7/197/211/195/2ØØ
/197/2Ø6}
44Ø POKE SP+218,2Ø1:POKE SP+219,199:
POKE SP+198,14Ø:SOUND 1,5,8,8
45Ø FOR X=48 TO 63 STEP Ø.Ø5:POKE 71
1,X:NEXT X:SOUND 1,Ø,Ø,Ø
46Ø POKE SP+198,138:POKE SP+218,8:PO
KE SP+219,7:POKE 711,54
47Ø ZS=ZS+ZA
48Ø PE=1ØØ-ZA:GP=GP+PE:POSITION 14,1
1:? #6;GP
49Ø IF DY=1Ø THEN 55Ø
5ØØ POSITION 5,11:? #6;"{3SPACES}"
51Ø POKE SP+218,Ø:POKE SP+219,Ø:POKE
 SP+21,196:POKE SP+22,198
52Ø READ A:POKE SP+A,194:SOUND 1,2Ø,
12,1Ø:FOR P=1 TO 2ØØ:NEXT P:SOUND 1,
Ø,Ø,Ø:POKE SP+A,66
53Ø POKE 2Ø,Ø:POKE 19,Ø
54Ø RETURN
549 REM {CODE2Ø6/197/213/197/16Ø/21Ø
/213/2Ø6/196/197}
55Ø RD=RD+1:IF DY<1Ø THEN 66Ø
56Ø IF ZS>6ØØ AND RD=5 THEN 135Ø
57Ø IF ZS>6ØØ THEN 66Ø
58Ø FOR P=1 TO 2ØØ:NEXT P
589 REM {CODE194/2Ø7/2Ø6/213/211}
59Ø BO=(6ØØ-ZS)*5Ø:POSITION Ø,9:? #6
;CHR$(129);"{CODE194/2Ø7/2Ø6/213/211
} ";BO;" {CODE2Ø8/213/2Ø6/2Ø3/212/19
7}";CHR$(129)
6ØØ GP=GP+BO:RESTORE 62Ø
61Ø FOR ML=1 TO 3Ø:READ TN,LE:SOUND
Ø,TN,1Ø,8:FOR P=1 TO LE*2Ø:NEXT P:SO
UND Ø,Ø,Ø,Ø:FOR Q=1 TO 1Ø:NEXT Q:NEX
T ML
62Ø DATA 1Ø8,2,81,1,81,1,72,1,64,1,8
1,1,64,1,72,1,1Ø8,1,81,1,81,1,72,1,6
4,1,81,3,85,2,1Ø8,1
63Ø DATA 81,1,81,1,72,1,64,1,6Ø,1,64
,1,72,1,81,1,85,1,1Ø8,1,96,1,85,1,81
,3,81,3
64Ø FOR P=1 TO 5ØØ:NEXT P
65Ø IF RD=5 THEN 135Ø
659 REM {CODE21Ø/213/2Ø6/196/197/2Ø6
/193/2Ø6/218/197/2Ø1/199/197}
66Ø GRAPHICS 18:POKE 756,ST/256
67Ø POSITION 4,1Ø:? #6;"RUNDE ";:FOR
 I=1 TO RD:? #6;CHR$(129);:NEXT I
68Ø FOR P=1 TO 5ØØ:NEXT P
69Ø GOSUB 1Ø6Ø
699 REM {CODE211/2Ø8/2Ø1/197/2Ø4/198
/197/2Ø4/196/197/21Ø}
7ØØ ON RD GOSUB 72Ø,76Ø,8ØØ,84Ø
71Ø POKE POS,ZUST:POKE SP+21,196:POK
E SP+22,198:DY=Ø:GOTO 3Ø
719 REM {CODE21Ø/213/2Ø6/196/197/16Ø
/177}
72Ø FOR I=1 TO 22:READ A:POKE SP+A,6
6:NEXT I:FOR I=1 TO 2Ø:READ A:POKE S
P+A,67:NEXT I:RETURN
73Ø DATA 28,43,5Ø,53,55,57,65,66,9Ø,
95,98,1Ø6,1Ø7,129,131,136,144,145,16
1,168,177,193
74Ø DATA 46,68,72,82,84,92,94,122,12
4,127,133,135,137,153,167,17Ø,175,18
4,185,192
75Ø DATA 173,47,165,83,86,113,74,3Ø,
1Ø8
759 REM {CODE21Ø/213/2Ø6/196/197/16Ø
/178}
76Ø POKE 7Ø9,168:FOR I=1 TO 23:READ
A:POKE SP+A,66:NEXT I:FOR I=1 TO 21:
READ A:POKE SP+A,67:NEXT I:RETURN
77Ø DATA 26,31,44,53,65,71,78,89,9Ø,
93,97,1Ø5,1Ø6,115,129,136,141,143,14
8,155,145,175,177
78Ø DATA 27,37,49,55,67,83,91,95,1Ø2
,111,126,128,137,144,152,166,17Ø,174
,182,188,192
79Ø DATA 87,47,62,147,19Ø,193,72,133
,75
799 REM {CODE21Ø/213/2Ø6/196/197/16Ø
/179}
8ØØ POKE 7Ø9,72:FOR I=1 TO 27:READ A
:POKE SP+A,66:NEXT I:FOR I=1 TO 19:R
EAD A:POKE SP+A,67:NEXT I:RETURN
81Ø DATA 47,51,55,64,68,72,76,88,94,
```

```
96,101,106,108,118,121,123,132,138,1
44,146,148,150,155,183,187,191,195
820 DATA 25,29,33,37,69,73,77,83,86,
90,105,114,131,151,154,161,165,169,1
73
830 DATA 126,31,82,75,70,171,107,134
,177
839 REM {CODE210/213/206/196/197/160
/180}
840 POKE 709,6:FOR I=1 TO 25:READ A:
POKE SP+A,66:NEXT I:FOR I=1 TO 23:RE
AD A:POKE SP+A,67:NEXT I:RETURN
850 DATA 31,34,45,49,58,64,68,74,86,
89,91,96,101,103,115,128,133,137,144
,146,170,175,181,186,192
860 DATA 27,44,47,52,56,61,69,72,83,
87,94,97,106,124,131,134,142,149,153
,182,185,188,195
870 DATA 109,66,173,75,132,166,143,1
38,176
879 REM {CODE212/201/212/197/204/194
/201/204/196}
880 GRAPHICS 18:POKE 756,ST/256
890 RD=0:GP=0
900 POSITION 4,0:? #6;"{CODE195/210/
193/218/217}{2SPACES}{CODE199/193/21
2/197/211}":POSITION 3,8:? #6;"COPYR
IGHT 4/84":POSITION 5,9:? #6;"AXEL{2
SPACES}BELZ"
910 POSITION 3,11:? #6;"druecke{2SPA
CES}{CODE243/244/225/242/244}"
920 FOR I=3 TO 15 STEP 2:POSITION I,
2:? #6;CHR$(3):POSITION I+1,2:? #6;C
HR$(2):NEXT I:POSITION 3,2:? #6;CHR$
(2)
930 FOR I=4 TO 15:POSITION I,4:? #6;
CHR$(33)
940 IF PEEK(53279)=6 OR STRIG(0)=0 T
HEN 550
950 SOUND 0,120,14,4:FOR P=1 TO 30:N
EXT P:SOUND 0,0,0,0
960 LOCATE I,2,TOR:IF TOR=2 THEN POS
ITION I,2:? #6;CHR$(3):GOTO 980
970 POSITION I,2:? #6;CHR$(2)
980 POSITION I,4:? #6;" ":NEXT I
990 FOR I=15 TO 4 STEP -1:POSITION I
,4:? #6;CHR$(161)
1000 IF PEEK(53279)=6 OR STRIG(0)=0
THEN 550
1010 SOUND 0,120,14,4:FOR P=1 TO 30:
NEXT P:SOUND 0,0,0,0
1020 LOCATE I,2,TOR:IF TOR=2 THEN PO
SITION I,2:? #6;CHR$(3):GOTO 1040
1030 POSITION I,2:? #6;CHR$(2)
1040 POSITION I,4:? #6;" ":NEXT I
1050 GOTO 930
1059 REM {CODE211/208/201/197/204/19
8/197/204/196}
1060 GRAPHICS 18:POKE 756,ST/256
1070 RESTORE 1080:FOR X=0 TO 4:READ
A:POKE 708+X,A:NEXT X
1080 DATA 235,120,183,54,0
1090 SP=PEEK(88)+256*PEEK(89):POS=SP
+197:ZUST=129
1100 FOR I=0 TO 19:POKE SP+I,66:NEXT
 I:FOR I=200 TO 217:POKE SP+I,66:NEX
T I
1110 FOR I=40 TO 180 STEP 20:POKE SP
+I,67:NEXT I:FOR I=39 TO 179 STEP 20
:POKE SP+I,67:NEXT I
1120 POKE SP+21,4:POKE SP+22,5:POKE
SP+198,138:POKE SP+199,139
1130 POSITION 0,11:? #6;"zeit{5SPACE
S}pkte ";GP
1140 RESTORE 690+RD*40:RETURN
1149 REM {CODE201/206/201/212/201/19
3/204/201/211/201/197/210/213/206/19
9}
1150 DIM CH$(32):RESTORE 1170
1160 FOR I=1 TO 32:READ A:CH$(I)=CHR
$(A):NEXT I
1170 DATA 104,104,133,204,104,133,20
3,104,133,206,104,133,205,162,4,160,
0
1180 DATA 177,203,145,205,136,208,24
9,230,204,230,206,202,208,240,96
1190 POKE 106,PEEK(106)-5:GRAPHICS 0
:ST=(PEEK(106)+1)*256:POKE 756,ST/25
6
1200 A=USR(ADR(CH$),57344,ST)
1210 RESTORE 1220:FOR I=ST+8 TO ST+1
11:READ A:POKE I,A:NEXT I:RETURN
1220 DATA 60,126,219,255,189,219,102
,60
1230 DATA 0,0,0,16,255,16,0,0
1240 DATA 8,8,8,8,28,8,8,8
1250 DATA 0,0,0,62,99,62,0,0
1260 DATA 0,0,0,0,224,0,0,0
1270 DATA 0,136,82,4,232,4,82,136
1280 DATA 0,0,0,124,198,124,0,0
1290 DATA 0,0,0,0,7,0,0,0
1300 DATA 0,17,74,32,23,32,74,17
1310 DATA 0,24,126,24,127,96,0,0
1320 DATA 0,0,0,0,255,0,0,0
1330 DATA 0,24,126,24,127,96,96,144
1340 DATA 18,213,98,26,233,78,209,50
1349 REM {CODE197/216/208/204/207/21
1/201/207/206/175/197/206/196/197}
1350 POSITION 0,0
1360 SOUND 2,75,8,15:V1=15:V2=15:V3=
15
1370 FOR I=1 TO 20:POKE 712,14
1380 ? #6;"{CODE13/13}--{CODE173}-{C
ODE173/141/13/141/173}";
```

```
1390 FOR P=1 TO 5:NEXT P:POKE 712,0:
NEXT I
1400 POSITION 0,11:? #6;"{19SPACES}"
:POSITION 3,11:? #6;GP;"{2SPACES}{CO
DE240/245/238/235/244/229}"
1410 IF DY=10 AND RD=5 AND ZUST=129
THEN POSITION 4,4:? #6;" SPIELENDE "
:GOTO 1430
1420 POSITION 1,4:? #6;" DAS WAR ZU
LANGE "
1430 POSITION 2,6:? #6;" druecke{2SP
ACES}start "
1440 DA=0.9
1450 SOUND 0,20,8,V1:SOUND 1,40,8,V2
:SOUND 2,70,8,V3
1460 V1=V1*DA:V2=V2*(DA+0.05):V3=V3*
(DA+0.08)
1470 IF V3>1 THEN 1450
1480 SOUND 0,0,0,0:SOUND 1,0,0,0:SOU
ND 2,0,0,0
1490 IF PEEK(53279)=6 OR STRIG(0)=0
THEN 20
1500 GOTO 1490
  READY !
```

Vier gewinnt

Beschreibung des Programmes (Atari 600XL/800XL, 6030 Bytes)
Dieses spielstarke und grafisch sehr schön aufgebaute Denkspiel stammt aus der HC-Diplom-Aktion unserer Schwesterzeitschrift HC-Mein Home-Computer. Gewonnen hat bei diesem Spiel, wer in einem Raster von sieben Schächten mit jeweils sechs Plätzen beim Auffüllen zuerst vier Chips in eine Reihe bekommt, egal ob horizontal, vertikal oder diagonal. Das Programm läßt Spiele gegen den Computer zu, aber auch Spiele zwischen zwei menschlichen Partnern oder Gefechte des Computers mit sich selbst.

Dietmar Specht

```
1 REM * VIER GEWINNT *
2 REM * FUER CHIP-SPECIAL *
3 REM * VON DIETMAR SPECHT *
4 REM
10 DIM NAME1$(8),NAME2$(8),NAME$(8)
20 DIM LINE(7)
30 FOR I=1 TO 7:LINE(I)=6:NEXT I
40 DIM TAB(7,6)
50 FOR I=0 TO 7:FOR J=0 TO 6:TAB(I,J
)=0:NEXT J:NEXT I
80 FOR I=1 TO 7:LINE(I)=6:NEXT I
90 GO=1
100 GRAPHICS 2
110 POSITION 8,5:? #6;CHR$(180):FOR
I=1 TO 50:NEXT I:POSITION 5,5:? #6;C
HR$(180)
115 FOR I=1 TO 99:NEXT I:POSITION 8,
4:? #6;CHR$(180):FOR I=1 TO 99:NEXT
I:POSITION 7,5:? #6;CHR$(180)
120 FOR I=1 TO 99:NEXT I:POSITION 8,
3:? #6;CHR$(180):FOR I=1 TO 99:NEXT
I:POSITION 8,2:? #6;CHR$(180)
125 FOR I=1 TO 99:NEXT I:POSITION 7,
4:? #6;CHR$(52):FOR I=1 TO 99:NEXT I
:POSITION 7,3:? #6;CHR$(180)
130 FOR I=1 TO 99:NEXT I:POSITION 6,
5:? #6;CHR$(180):FOR I=1 TO 99:NEXT
I:POSITION 6,4:? #6;CHR$(180)
135 POSITION 10,5:? #6;CHR$(103);CHR
$(101);CHR$(119);CHR$(105);CHR$(110)
;CHR$(110);CHR$(116);CHR$(161)
138 FOR I=1 TO 300:NEXT I
139 ? :? "COMPUTER ALS SPIELER IST *
":?
140 POSITION 0,8:? #6;"NAME {CODE177
/174} SPIELER:":INPUT NAME1$
150 POSITION 0,9:? #6;CHR$(110);CHR$
(97);CHR$(109);CHR$(101);" ";CHR$(17
8);CHR$(174);" ";
152 ? #6;CHR$(115);CHR$(112);CHR$(10
5);CHR$(101);CHR$(108);CHR$(101);CHR
$(114);CHR$(58)
160 INPUT NAME2$
300 GRAPHICS 7
305 POKE 559,0
310 COLOR 3
315 GOSUB 10010
320 SETCOLOR 4,0,7
325 POKE 559,34
330 PLOT 42,9:DRAWTO 42,70:DRAWTO 15
```

```
7,70:DRAWTO 157,9:DRAWTO 43,9:DRAWTO
 43,69
340 PLOT 156,9:DRAWTO 156,70
350 FOR Y=10 TO 60 STEP 10
360 FOR X=44 TO 140 STEP 16
370 PLOT X,Y:DRAWTO X+15,Y:DRAWTO X+
15,Y+9:DRAWTO X,Y+9:DRAWTO X,Y
380 PLOT X+1,Y+1:DRAWTO X+1,Y+8
390 PLOT X+14,Y+1:DRAWTO X+14,Y+8
400 NEXT X:NEXT Y
410 FOR C=1 TO 2
420 COLOR C:X=(C-1)*14:XP11=X+11
430 FOR Y=9 TO 69 STEP 3
440 PLOT X,Y:DRAWTO XP11,Y
450 PLOT X,Y+1:DRAWTO XP11,Y+1
460 NEXT Y:NEXT C
465 SETCOLOR 3,0,0
470 POKE 87,1
480 FOR I=1 TO 7:POSITION 4+2*I,1:?
#6;CHR$(144+I):NEXT I
490 POKE 87,7
500 IF GO=1 THEN NAME$=NAME1$
510 IF GO=2 THEN NAME$=NAME2$
520 GOSUB 700
530 IF NAME$="*" THEN GOTO 4000
540 ? "DEIN ZUG ";NAME$
550 ?
560 TRAP 620
570 INPUT COLNO
580 IF COLNO<1 THEN GOTO 620
590 IF COLNO>7 THEN GOTO 620
600 IF LINE(COLNO)=0 THEN GOTO 620
610 TRAP 40000
615 GOTO 1000
620 ? "FEHLER! BITTE 1 BIS 7, ";NAME
$;"!"
630 GOTO 550
700 Y=9+CT*3
710 X=(GO-1)*14:XP11=X+11
720 FOR V=Y-1 TO 7 STEP -1
730 COLOR GO:PLOT X,V:DRAWTO XP11,V
740 COLOR 0:PLOT X,V+2:DRAWTO XP11,V
+2
750 NEXT V
760 COLOR GO
770 FOR V=6 TO 1 STEP -1
780 PLOT X,V:DRAWTO X+11,V
790 NEXT V
800 FOR V=X TO 28
810 COLOR GO:PLOT V+12,1:DRAWTO V+12
,8
820 COLOR 0:PLOT V,1:DRAWTO V,8
830 NEXT V
840 RETURN
1000 LINE(COLNO)=LINE(COLNO)-1
1010 Y=11+LINE(COLNO)*10
1020 X=(COLNO-1)*16+46
1030 FOR V=28 TO X-1
1040 COLOR GO:PLOT V+12,1:DRAWTO V+1
2,8
1050 COLOR 0:PLOT V,1:DRAWTO V,8
1060 NEXT V
1070 FOR V=1 TO Y-1:VP8=V+8
1080 I=VP8-10*INT(VP8/10):IF I<>0 AN
D I<>9 THEN COLOR GO:PLOT X,VP8:DRAW
TO X+11,VP8
1090 I=V-10*INT(V/10):IF I<>0 AND I<
>9 THEN COLOR 0:PLOT X,V:DRAWTO X+11
,V
1100 NEXT V
1190 TAB(COLNO,LINE(COLNO))=GO
1200 WIN=0
1210 GOSUB 2000
1220 IF WIN=1 THEN GOTO 1500
1300 IF GO=1 THEN GO=2:GOTO 500
1310 IF CT=20 THEN ? "UNENTSCHIEDEN
- KEINER GEWINNT":GOTO 1510
1320 GO=1:CT=CT+1:GOTO 500
1500 ? NAME$;" IST DER SIEGER"
1510 ? :? "NOCH EIN SPIEL?J/N";
1520 INPUT NAME$
1530 IF NAME$<>"J" AND NAME$<>"N" TH
EN GOTO 1510
1540 IF NAME$="J" THEN RUN
1550 GRAPHICS 2+16
1560 ? #6;"AUF WIEDERSEHEN":? #6:? #
6;"bis zum":? #6;"naechsten mal"
1570 FOR I=1 TO 600:NEXT I:END
2000 X=COLNO
2010 Y=LINE(X)
2020 NO=1
2100 X=X-1
2110 IF X=0 THEN GOTO 2200
2120 SQ=TAB(X,Y)
2130 IF SQ<>GO THEN GOTO 2200
2150 NO=NO+1
2160 IF NO=4 THEN WIN=1:RETURN
2170 GOTO 2100
2200 X=COLNO
2300 X=X+1
2310 IF X=8 THEN GOTO 2400
2320 SQ=TAB(X,Y)
2330 IF SQ<>GO THEN GOTO 2400
2350 NO=NO+1
2360 IF NO=4 THEN WIN=1:RETURN
2370 GOTO 2300
2400 X=COLNO:Y=LINE(X):NO=1
2500 Y=Y+1
2510 IF Y=6 THEN GOTO 2600
2520 SQ=TAB(X,Y)
2530 IF SQ<>GO THEN GOTO 2600
2540 NO=NO+1
2550 IF NO=4 THEN WIN=1:RETURN
2560 GOTO 2500
```

```
2600 X=COLNO:Y=LINE(X):NO=1
2700 Y=Y+1:X=X-1
2710 IF X=0 OR Y=6 THEN GOTO 2800
2720 SQ=TAB(X,Y)
2730 IF SQ<>GO THEN GOTO 2800
2740 NO=NO+1
2750 IF NO=4 THEN WIN=1:RETURN
2760 GOTO 2700
2800 X=COLNO:Y=LINE(X)
2900 Y=Y-1:X=X+1
2910 IF X=8 OR Y<0 THEN GOTO 3000
2920 SQ=TAB(X,Y)
2930 IF SQ<>GO THEN GOTO 3000
2940 NO=NO+1
2950 IF NO=4 THEN WIN=1:RETURN
2960 GOTO 2900
3000 X=COLNO:Y=LINE(X):NO=1
3100 Y=Y-1:X=X-1
3110 IF X=0 OR Y<0 THEN GOTO 3200
3120 SQ=TAB(X,Y)
3130 IF SQ<>GO THEN GOTO 3200
3140 NO=NO+1
3150 IF NO=4 THEN WIN=1:RETURN
3160 GOTO 3100
3200 X=COLNO:Y=LINE(X)
3300 Y=Y+1:X=X+1
3310 IF X=8 OR Y=6 THEN GOTO 3400
3320 SQ=TAB(X,Y)
3330 IF SQ<>GO THEN GOTO 3400
3340 NO=NO+1
3350 IF NO=4 THEN WIN=1:RETURN
3360 GOTO 3300
3400 RETURN
4000 COLNO=1
4002 IF LINE(COLNO)=0 THEN 4030
4004 LINE(COLNO)=LINE(COLNO)-1
4006 TAB(COLNO,LINE(COLNO))=GO
4010 GOSUB 2000
4016 TAB(COLNO,LINE(COLNO))=0
4018 LINE(COLNO)=LINE(COLNO)+1
4020 IF WIN=1 THEN GOTO 1000
4030 COLNO=COLNO+1
4040 IF COLNO<8 THEN GOTO 4002
4100 STCOL=GO
4200 IF GO=1 THEN GO=2:GOTO 4220
4210 GO=1
4220 COLNO=1
4222 IF LINE(COLNO)=0 THEN GOTO 4250
4224 LINE(COLNO)=LINE(COLNO)-1
4226 TAB(COLNO,LINE(COLNO))=GO
4230 GOSUB 2000
4236 TAB(COLNO,LINE(COLNO))=0
4238 LINE(COLNO)=LINE(COLNO)+1
4240 IF WIN=1 THEN GO=STCOL:GOTO 100
0
4250 COLNO=COLNO+1
4260 IF COLNO<8 THEN GOTO 4222
4300 GO=STCOL
4400 COLNO=INT(RND(1)*7)+1
4410 IF LINE(COLNO)=0 THEN GOTO 4400
4500 GOTO 1000
5000 END
10010 DLIST=PEEK(560)+PEEK(561)*256+
4
10020 POKE DLIST-1,70:POKE DLIST+2,6
10030 I=3
10040 DLHI=PEEK(DLIST+I+6)
10045 RETURN
10050 POKE DLIST+I,PEEK(DLIST+I+6)
10060 IF DLHI=PEEK(561) THEN RETURN
10070 I=I+1
10080 IF I>100 THEN STOP
10090 GOTO 10040
  READY !
```

Change

Beschreibung des Programmes (Atari 600XL/800XL, 10100 Bytes)
Hilfreich ist dieses Programm für jeden, der oft Geldbeträge in fremde Währungen umrechnet. Sei es beruflich im Import und Export oder privat für die Urlaubsreise. Man gibt einfach den Geldbetrag und anschließend den Kennbuchstaben für die Währungseinheit dieses Betrages ein. Der Computer rechnet den Betrag in die anderen 21 Währungen um und gibt eine Tabelle mit den Geldbeträgen, den Währungseinheiten und den Landeskennern aus. Diese Tabelle ist für eine Bildschirmseite zu umfangreich und wird deshalb auf zwei Seiten, zwischen denen man durch Tastenbdruck hin- und herschalten kann, verteilt. Nachdem das Programm mit der Anweisung RUN und dem Tastendruck RETURN gestartet wurde, können neue Umrechnungskurse im Dialog eingegeben werden. Beachten Sie bitte, daß die Kurse mit einem Dezimalpunkt und nicht, wie im deutschen Sprachgebiet üblich, mit einem Dezimalkomma einge-

tippt werden. Es ist nicht notwendig, bereits beim ersten Programmdurchlauf neue Umrechnungskurse einzugeben.

Willibald Rauffer

```
10 REM * WAEHRUNG *
11 REM * FUER CHIP-SPECIAL *
12 REM * VON W.RAUFFER *
13 REM
30 REM {CODE21/21/21/21/21/21/21} We
rtzuweisungen {CODE21/21/21/21/21/21
}
40 REM
50 DIM A$(20):LET A$=" Mark{10SPACES
}(D){2SPACES}"
60 DIM B$(20):LET B$=" Dollar{8SPACE
S}(USA)"
70 DIM C$(20):LET C$=" Engl. Pfund{3
SPACES}(GB) "
80 DIM D$(20):LET D$=" Can. Dollar{3
SPACES}(CDN)"
90 DIM E$(20):LET E$=" Gulden{8SPACE
S}(NL) "
100 DIM F$(20):LET F$=" Franken{7SPA
CES}(CH) "
110 DIM G$(20):LET G$=" Belg. Franc{
3SPACES}(B){2SPACES}"
120 DIM H$(20):LET H$=" Franz. Franc
{2SPACES}(F){2SPACES}"
130 DIM I$(20):LET I$=" D{CODE11}n.
Krone{4SPACES}(DK) "
140 DIM J$(20):LET J$=" Norw. Krone{
3SPACES}(N){2SPACES}"
150 DIM K$(20):LET K$=" Schwed. Kron
e (S){2SPACES}"
160 DIM L$(20):LET L$=" Lira{10SPACE
S}(I){2SPACES}"
170 DIM M$(20):LET M$=" Schilling{5S
PACES}(A){2SPACES}"
180 DIM N$(20):LET N$=" Peseta{8SPAC
ES}(E){2SPACES}"
190 DIM O$(20):LET O$=" Escudo{8SPAC
ES}(P){2SPACES}"
200 DIM P$(20):LET P$=" Yen{11SPACES
}(J){2SPACES}"
210 DIM Q$(20):LET Q$=" Dinar{9SPACE
S}(YU) "
220 DIM R$(20):LET R$=" Drachme{7SPA
CES}(GR) "
230 DIM S$(20):LET S$=" T{CODE10}rk.
 Pfund{3SPACES}(TR) "
240 DIM T$(20):LET T$=" Finnmark{6SP
ACES}(SF) "
250 DIM U$(20):LET U$=" Ir. Pfund{5S
PACES}(IRL)"
260 DIM V$(20):LET V$=" Ostmark{7SPA
CES}(DDR)"
300 DIM A(21)
310 DIM Z$(40):LET Z$="{40SPACES}"
320 DIM F(21)
330 LET A(1)=2.77:LET A(2)=3.86:LET
A(3)=2.16:LET A(4)=0.8975:LET A(5)=1
.225:LET A(6)=0.0496:LET A(7)=0.335
340 LET A(8)=0.2825:LET A(9)=0.36:LE
T A(10)=0.3475:LET A(11)=1.68E-03:LE
T A(12)=0.1435:LET A(13)=0.0188
350 LET A(14)=0.0245:LET A(15)=0.012
1:LET A(16)=0.024:LET A(17)=0.03:LET
 A(18)=0.011:LET A(19)=0.48:LET A(20
)=3.15
360 LET A(21)=0.205
370 DIM X$(1)
380 REM
500 REM {CODE21/21/21/21/21/21/21/21
/21/21} Seite 1 {CODE21/21/21/21/21/
21/21/21/21/21}
510 REM
520 REM Der Inhalt von Bildschirmsei
t 1 wird aufgebaut.
540 REM
550 PRINT CHR$(125):GOSUB 5000
560 POSITION 4,7:PRINT "Das Programm
 W{CODE11}hrung rechnet die":POSITIO
N 4,9:PRINT "W{CODE11}hrungen der wi
chtigsten{2SPACES}euro-"
570 POSITION 4,11:PRINT "p{CODE11}is
chen und{2SPACES}aussereurop{CODE11}
ischen":POSITION 9,13:PRINT "L{CODE1
1}nder ineinander um."
580 POSITION 4,21:PRINT "Dr{CODE10}c
ken Sie bitte die{2SPACES}Taste mit"
:POSITION 4,22:PRINT "der Aufschrift
 W und dann RETURN"
590 DIM W$(1)
600 INPUT W$
610 IF W$="W" THEN GOTO 800
620 GOTO 600
630 REM
800 REM {CODE21/21/21/21/21/21/21/21
/21/21} Seite 2 {CODE21/21/21/21/21/
21/21/21/21/21}
810 REM
820 REM Der Inhalt von Bildschirmsei
t 2 wird aufgebaut.
840 REM
850 PRINT CHR$(125):GOSUB 5000
860 POSITION 4,6:PRINT "Wollen Sie d
ie{2SPACES}bisher abgespei-":POSITIO
N 4,8:PRINT "cherten Umrechnungskurs
e verwen-"
870 POSITION 4,10:PRINT "den, oder m
{CODE15}chten{2SPACES}Sie neue{2SPAC
ES}Um-":POSITION 8,12:PRINT "rechnun
```

```
gskurse eingeben?"
88Ø POSITION 13,15:PRINT "1{2SPACES}
alte Kurse":POSITION 13,17:PRINT "2{
2SPACES}neue Kurse"
89Ø POSITION 4,2Ø:PRINT "Dr{CODE1Ø}c
ken Sie bitte die{2SPACES}Taste mit"
:POSITION 4,21:PRINT "der{2SPACES}Au
fschrift{2SPACES}1{2SPACES}oder{2SPA
CES}2, und"
9ØØ POSITION 14,22:PRINT "dann RETUR
N"
91Ø INPUT W
92Ø IF W=1 THEN GOTO 18ØØ
93Ø IF W=2 THEN GOTO 1ØØØ
94Ø GOTO 91Ø
95Ø REM
1ØØØ REM {CODE21/21/21/21/21/21/21/2
1/21/21} Seite 3 {CODE21/21/21/21/21
/21/21/21/21}
1Ø1Ø REM
1Ø2Ø REM Hier{2SPACES}werden die neu
en{2SPACES}Kurse
1Ø3Ø REM eingegeben.
1Ø35 REM
1Ø4Ø REM
1Ø45 POSITION 4,14:PRINT Z$:POSITION
 4,16:PRINT Z$
1Ø5Ø GOSUB 51ØØ:PRINT B$;:GOSUB 52ØØ
1Ø6Ø PRINT A(1):GOSUB 53ØØ
1Ø7Ø LET A(1)=B
1Ø8Ø GOSUB 51ØØ:PRINT C$;:GOSUB 52ØØ
1Ø9Ø PRINT A(2):GOSUB 53ØØ
11ØØ LET A(2)=B
111Ø GOSUB 51ØØ:PRINT D$;:GOSUB 52ØØ
112Ø PRINT A(3):GOSUB 53ØØ
113Ø LET A(3)=B
114Ø GOSUB 51ØØ:PRINT E$;:GOSUB 52ØØ
115Ø PRINT A(4):GOSUB 53ØØ
116Ø LET A(4)=B
117Ø GOSUB 51ØØ:PRINT F$;:GOSUB 52ØØ
118Ø PRINT A(5):GOSUB 53ØØ
119Ø LET A(5)=B
12ØØ GOSUB 51ØØ:PRINT G$;:GOSUB 52ØØ
121Ø PRINT A(6):GOSUB 53ØØ
122Ø LET A(6)=B
123Ø GOSUB 51ØØ:PRINT H$;:GOSUB 52ØØ
124Ø PRINT A(7):GOSUB 53ØØ
125Ø LET A(7)=B
126Ø GOSUB 51ØØ:PRINT I$;:GOSUB 52ØØ
127Ø PRINT A(8):GOSUB 53ØØ
128Ø LET A(8)=B
129Ø GOSUB 51ØØ:PRINT J$;:GOSUB 52ØØ
13ØØ PRINT A(9):GOSUB 53ØØ
131Ø LET A(9)=B
132Ø GOSUB 51ØØ:PRINT K$;:GOSUB 52ØØ
133Ø PRINT A(1Ø):GOSUB 53ØØ
134Ø LET A(1Ø)=B
135Ø GOSUB 51ØØ:PRINT L$;:GOSUB 52ØØ
136Ø PRINT A(11):GOSUB 53ØØ
137Ø LET A(11)=B
138Ø GOSUB 51ØØ:PRINT M$;:GOSUB 52ØØ
139Ø PRINT A(12):GOSUB 53ØØ
14ØØ LET A(12)=B
141Ø GOSUB 51ØØ:PRINT N$;:GOSUB 52ØØ
142Ø PRINT A(13):GOSUB 53ØØ
143Ø LET A(13)=B
144Ø GOSUB 51ØØ:PRINT O$;:GOSUB 52ØØ
145Ø PRINT A(14):GOSUB 53ØØ
146Ø LET A(14)=B
147Ø GOSUB 51ØØ:PRINT P$;:GOSUB 52ØØ
148Ø PRINT A(15):GOSUB 53ØØ
149Ø LET A(15)=B
15ØØ GOSUB 51ØØ:PRINT Q$;:GOSUB 52ØØ
151Ø PRINT A(16):GOSUB 53ØØ
152Ø LET A(16)=B
153Ø GOSUB 51ØØ:PRINT R$;:GOSUB 52ØØ
154Ø PRINT A(17):GOSUB 53ØØ
155Ø LET A(17)=B
156Ø GOSUB 51ØØ:PRINT S$;:GOSUB 52ØØ
157Ø PRINT A(18):GOSUB 53ØØ
158Ø LET A(18)=B
159Ø GOSUB 51ØØ:PRINT T$;:GOSUB 52ØØ
16ØØ PRINT A(19):GOSUB 53ØØ
161Ø LET A(19)=B
162Ø GOSUB 51ØØ:PRINT U$;:GOSUB 52ØØ
163Ø PRINT A(2Ø):GOSUB 53ØØ
164Ø LET A(2Ø)=B
165Ø GOSUB 51ØØ:PRINT V$;:GOSUB 52ØØ
166Ø PRINT A(21):GOSUB 53ØØ
167Ø LET A(21)=B
168Ø REM
18ØØ REM {CODE21/21/21/21/21/21/21/2
1/21/21} Seite 4 {CODE21/21/21/21/21
/21/21/21/21}
181Ø REM
182Ø REM Erster Teil der{2SPACES}Ein
gabelist
183Ø REM
184Ø PRINT CHR$(125)
185Ø POSITION 4,1:PRINT "Betrag = ?"
186Ø GOSUB 54ØØ
187Ø POSITION 4,18:PRINT "Geben Sie{
2SPACES}bitte den{2SPACES}Betrag ein
,":POSITION 4,19:PRINT "dr{CODE1Ø}ck
en Sie dann die RETURN-Taste"
188Ø INPUT X
189Ø POSITION 13,1:PRINT X
19ØØ GOSUB 55ØØ:GOSUB 57ØØ
191Ø INPUT C
192Ø IF C=1 THEN GOTO 2ØØØ
193Ø IF C=2 THEN GOTO 3ØØØ
194Ø GOTO 191Ø
195Ø REM
2ØØØ REM {CODE21/21/21/21/21/21/21/2
```

```
1/21/21} Seite 5 {CODE21/21/21/21/21
/21/21/21/21}
2Ø1Ø REM
2Ø2Ø REM Zweiter Teil der Eingabelis
t
2Ø3Ø REM
2Ø4Ø PRINT CHR$(125)
2Ø5Ø POSITION 4,2:PRINT "W{CODE11}hr
ung:"
2Ø6Ø POSITION 9,4:PRINT "K ";K$
2Ø7Ø POSITION 9,5:PRINT "L ";L$
2Ø8Ø POSITION 9,6:PRINT "M ";M$
2Ø9Ø POSITION 9,7:PRINT "N ";N$
21ØØ POSITION 9,8:PRINT "O ";O$
211Ø POSITION 9,9:PRINT "P ";P$
212Ø POSITION 9,1Ø:PRINT "Q ";Q$
213Ø POSITION 9,11:PRINT "R ";R$
214Ø POSITION 9,12:PRINT "S ";S$
215Ø POSITION 9,13:PRINT "T ";T$
216Ø POSITION 9,14:PRINT "U ";U$
217Ø POSITION 9,15:PRINT "V ";V$
218Ø GOSUB 56ØØ:GOSUB 57ØØ
219Ø INPUT C
22ØØ IF C=1 THEN GOTO 25ØØ
221Ø IF C=2 THEN GOTO 3ØØØ
222Ø GOTO 219Ø
223Ø REM
25ØØ REM {CODE21/21/21/21/21} Seite
4 mit Betrag {CODE21/21/21}
251Ø REM
252Ø PRINT CHR$(125)
253Ø POSITION 4,1:PRINT "Betrag = ";
254Ø PRINT X
255Ø GOSUB 54ØØ:GOSUB 55ØØ:GOSUB 57Ø
Ø
256Ø GOTO 191Ø
257Ø REM
3ØØØ REM {CODE21/21/21/21/21} Waehru
ng eingeben {CODE21/21/21/21}
3Ø1Ø REM
3Ø2Ø REM {CODE15/15/15}Zeilen 18-22
loeschen {CODE15/15/15}
3Ø3Ø REM
3Ø4Ø FOR Y=17 TO 22:POSITION Ø,Y
3Ø5Ø PRINT Z$:NEXT Y
3Ø6Ø REM
3Ø7Ø REM {CODE15/15/15/15/15/15/15/1
5/15/15/15/15/15/15/15/15/15/15/15/1
5/15/15/15/15/15/15/15}
3Ø8Ø REM
3Ø9Ø POSITION 4,18:PRINT "Geben Sie
bitte den Kennbuchsta-":POSITION 4,1
9:PRINT "ben f{CODE1Ø}r die gew{CODE
1Ø}nschte W{CODE11}hrungs-"
31ØØ POSITION 4,2Ø:PRINT "einheit ei
n,{2SPACES}dr{CODE1Ø}cken{2SPACES}Si
e{2SPACES}dann":POSITION 12,21:PRINT
"die RETURN-Taste"
311Ø INPUT X$
312Ø REM
35ØØ REM {CODE21/21/21/21} Waehrunge
n umrechnen {CODE21/21
351Ø REM
352Ø LET E=(X$="A")*X+(X$="B")*X*A(1
)+(X$="C")*X*A(2)+(X$="D")*X*A(3)+(X
$="E")*X*A(4)+(X$="F")*X*A(5)
353Ø LET E=E+(X$="G")*X*A(6)+(X$="H"
)*X*A(7)+(X$="I")*X*A(8)+(X$="J")*X*
A(9)+(X$="K")*X*A(1Ø)
354Ø LET E=E+(X$="L")*X*A(11)+(X$="M
")*X*A(12)+(X$="N")*X*A(13)+(X$="O")
*X*A(14)+(X$="P")*X*A(15)
355Ø LET E=E+(X$="Q")*X*A(16)+(X$="R
")*X*A(17)+(X$="S")*X*A(18)+(X$="T")
*X*A(19)+(X$="U")*X*A(2Ø)
356Ø LET E=E+(X$="V")*X*A(21)
357Ø FOR I=1 TO 21
358Ø LET F(I)=E/A(I)
359Ø NEXT I
36ØØ REM
4ØØØ REM {CODE21/21/21/21/21/21/21/2
1/21/21} Seite 6 {CODE21/21/21/21/21
/21/21/21/21}
4Ø1Ø REM
4Ø2Ø REM Erster Teil der{2SPACES}Aus
gabelist
4Ø3Ø REM
4Ø4Ø PRINT CHR$(125):GOSUB 5ØØØ
4Ø5Ø POSITION Ø,5:PRINT E:POSITION 2
Ø,5:PRINT A$:POSITION Ø,6:PRINT F(1)
:POSITION 2Ø,6:PRINT B$
4Ø6Ø POSITION Ø,7:PRINT F(2):POSITIO
N 2Ø,7:PRINT C$:POSITION Ø,8:PRINT F
(3):POSITION 2Ø,8:PRINT D$
4Ø7Ø POSITION Ø,9:PRINT F(4):POSITIO
N 2Ø,9:PRINT E$:POSITION Ø,1Ø:PRINT
F(5):POSITION 2Ø,1Ø:PRINT F$
4Ø8Ø POSITION Ø,11:PRINT F(6):POSITI
ON 2Ø,11:PRINT G$:POSITION Ø,12:PRIN
T F(7):POSITION 2Ø,12:PRINT H$
4Ø9Ø POSITION Ø,13:PRINT F(8):POSITI
ON 2Ø,13:PRINT I$:POSITION Ø,14:PRIN
T F(9):POSITION 2Ø,14:PRINT J$
41ØØ POSITION Ø,15:PRINT F(1Ø):POSIT
ION 2Ø,15:PRINT K$
411Ø GOSUB 55ØØ
412Ø INPUT C
413Ø IF C=1 THEN GOTO 45ØØ
414Ø GOTO 412Ø
415Ø REM
45ØØ REM {CODE21/21/21/21/21/21/21/2
1/21/21} Seite 7 {CODE21/21/21/21/21
/21/21/21/21}
451Ø REM
```

```
4520 REM Zweiter Teil der Ausgabelis
t
4530 REM
4540 PRINT CHR$(125):GOSUB 5000
4550 POSITION 0,5:PRINT F(11):POSITI
ON 20,5:PRINT L$:POSITION 0,6:PRINT
F(12):POSITION 20,6:PRINT M$
4560 POSITION 0,7:PRINT F(13):POSITI
ON 20,7:PRINT N$:POSITION 0,8:PRINT
F(14):POSITION 20,8:PRINT O$
4570 POSITION 0,9:PRINT F(15):POSITI
ON 20,9:PRINT P$:POSITION 0,10:PRINT
 F(16):POSITION 20,10:PRINT Q$
4580 POSITION 0,11:PRINT F(17):POSIT
ION 20,11:PRINT R$:POSITION 0,12:PRI
NT F(18):POSITION 20,12:PRINT S$
4590 POSITION 0,13:PRINT F(19):POSIT
ION 20,13:PRINT T$:POSITION 0,14:PRI
NT F(20):POSITION 20,14:PRINT U$
4600 POSITION 0,15:PRINT F(21):POSIT
ION 20,15:PRINT V$
4610 GOSUB 5600
4620 POSITION 4,21:PRINT "Nocheinmal
? Ja oder Nein? Dr{CODE10}k-":POSITI
ON 4,22:PRINT "ken Sie bitte J oder
N,{2SPACES}RETURN"
4630 INPUT W$
4640 IF W$="1" THEN GOTO 4000
4650 IF W$="J" THEN GOTO 800
4660 IF W$="N" THEN END
4670 REM
5000 REM {CODE21/21/21/21/21/21} Unt
erprogramm 1 {CODE21/21/21/21/21}
5010 REM
5020 POKE 756,204:POSITION 13,1:PRIN
T "W {CODE11} h r u n g":POSITION 13
,0:PRINT "============":POSITION 13
,2
5030 PRINT "============":RETURN
5040 REM
5100 REM {CODE21/21/21/21/21/21} Unt
erprogramm 2 {CODE21/21/21/21/21}
5110 REM
5120 PRINT CHR$(125):GOSUB 5000:POSI
TION 9,6:RETURN
5130 REM
5200 REM {CODE21/21/21/21/21/21} Unt
erprogramm 3 {CODE21/21/21/21/21}
5210 REM
5220 PRINT ":":POSITION 6,10:PRINT "
alter Umrechnungskurs = ";:RETURN
5230 REM
5300 REM {CODE21/21/21/21/21/21} Unt
erprogramm 4 {CODE21/21/21/21/21}
5310 REM
5320 POSITION 6,13:PRINT "neuer Umre
chnungskurs = ?"
5330 POSITION 4,19:PRINT "Geben Sie
bitte den neuen Umrech-":POSITION 4,
20:PRINT "nungskurs{2SPACES}ein, und
{2SPACES}dr{CODE10}cken{2SPACES}Sie"
5340 POSITION 10,21:PRINT "dann die
RETURN-Taste":INPUT B:RETURN
5350 REM
5400 REM {CODE21/21/21/21/21/21} Unt
erprogramm 5 {CODE21/21/21/21/21}
5410 REM
5420 REM Waehrungsliste auf Seite{2S
PACES}4
5430 REM
5440 POSITION 4,4:PRINT "W{CODE11}hr
ung:":POSITION 9,6:PRINT "A ";A$:POS
ITION 9,7:PRINT "B ";B$
5450 POSITION 9,8:PRINT "C ";C$:POSI
TION 9,9:PRINT "D ";D$:POSITION 9,10
:PRINT "E ";E$
5460 POSITION 9,11:PRINT "F ";F$:POS
ITION 9,12:PRINT "G ";G$:POSITION 9,
13:PRINT "H ";H$
5470 POSITION 9,14:PRINT "I ";I$:POS
ITION 9,15:PRINT "J ";J$
5480 RETURN
5490 REM
5500 REM {CODE21/21/21/21/21/21} Unt
erprogramm 6 {CODE21/21/21/21/21}
5510 REM
5520 POSITION 4,18:PRINT "Dr{CODE10}
cken Sie bitte 1 und RETURN,{2SPACES
}":POSITION 4,19
5530 PRINT "{3SPACES}um die Liste fo
rtzusetzen{5SPACES}"
5540 POSITION 0,20:PRINT Z$
5550 RETURN
5560 REM
5600 REM {CODE21/21/21/21/21/21} Unt
erprogramm 7 {CODE21/21/21/21/21}
5610 REM
5620 POSITION 4,18:PRINT "Dr{CODE10}
cken Sie bitte 1 und RETURN,":POSITI
ON 6,19:PRINT "um die Liste zur{CODE
10}ckzusetzen"
5630 RETURN
5640 REM
5700 REM {CODE21/21/21/21/21/21} Unt
erprogramm 8 {CODE21/21/21/21/21}
5710 REM
5720 POSITION 4,21:PRINT "Dr{CODE10}
cken Sie bitte 2 und RETURN,":POSITI
ON 7,22:PRINT "um die W{CODE11}hrung
 einzugeben"
5730 RETURN

READY !
```

Player-Missile-Graphik

Ein Schnellkursus für Einsteiger mit Basic-Kenntnissen.
Teil 1: Der Player
Player-Missile Graphik, was ist das eigentlich? Übersetzt heißt es soviel wie Spieler-Geschoß-Graphik, und das sagt schon eine ganze Menge über die Anwendungsmöglichkeiten aus. Die PM-Graphik erlaubt nämlich eine fließende Bewegung von bis zu acht unabhängigen Objekten mit annehmbarer Geschwindigkeit, auch in Basic. Sie gibt dem Basic-Programmierer damit eine Möglichkeit in die Hand, auf relativ einfache Weise selbst Spiele mit bewegter Graphik zu erstellen. Bevor man nun jedoch darangehen kann, seinen ersten Player zu schaffen, muß man sich erst einmal ein paar grundsätzliche Gedanken dazu machen, und zwar:

<u>Wie soll der Player, die Spielfigur, aussehen?</u>
Man nehme sich einen Bleistift und etwas gerastertes Papier. Auf diesem Papier zählt man jetzt acht Kästchen ab und zieht vor dem ersten und letzen Kästchen einen vertikalen Strich über das ganze Papier. Man erhält einen Streifen, der dem Player-Band im Computer gleicht. Die acht Kästchen entsprechen dabei der acht Bit Breite eines Players. Ein Bit kann den Zustand 0=Nein oder 1=Ja beinhalten. Bei Ja, also 1, ist ein Punkt auf dem Bildschirm im Playerband zu sehen, bei Nein, also 0, logischerweise nicht.
Jetzt muß unser Player (Spielerbild) in den Streifen gebracht werden. Damit es am Anfang etwas einfacher wird, benutzen wir als Player ein simples Kreuz.

```
 OO
OOOO
 OO
```

Zweierpotenzen (2^n)

```
Die Bit-     7 6 5 4 3 2 1 0
folge ist:   0-0-0-1-1-0-0-0
             0-0-1-1-1-1-0-0
             0-0-0-1-1-0-0-0
```

Dieses Bitmuster entspricht der Darstellungsweise im binären Zahlensystem, welches der Computer intern verarbeitet. Sie muß nun umgerechnet werden in eine Dezimalschreibweise, also das Zahlensystem, mit dem wir rechnen. Die folgende Tabelle zeigt Zusammenhänge zwischen Dezimal- und binärem Zahlensystem auf. Hier die Bitfolge:

```
--- --- --- --- --- --- --- ---
 7   6   5   4   3   2   1   0
--- --- --- --- --- --- --- ---

--------- 8 Bit = 1 Byte -------
```

$2^7=128$ $2^6=64$ $2^5=32$ $2^4=16$
$2^3=8$ $2^2=4$ $2^1=2$ $2^0=1$

Daraus ergeben sich für unseren Player folgende Werte:

2^4+2^3 = 16+8 = 24 Dezimal

$2^5+2^4+2^3+2^2$ = 32+16+8+4= 60 Dezimal

2^4+2^3 = 16+8 = 24 Dezimal

<u>In welchen Speicherbereich kann ich die PM-Graphik bringen?</u>
Die Player-Missile-Basisadresse liegt bei 54279 Dezimal. In dieser Adresse wird nun hinterlegt, wieviel Speicherplatz für die PM-Graphik benötigt wird. In der Adresse 106 Dezimal befindet sich die Angabe des höchsten Speicherbereichs in Pages (Seiten), wobei eine Seite 256 Bytes lang ist. Wenn man nun ausreichend Speicherplatz für die PM-Graphik reservieren will, poked man einfach den Wert von PEEK (106), (also die Anzahl der Seiten) abzüglich des Speicherplatzes für die PM-Graphik in Adresse 54279. Für unser Beispiel sieht das dann so aus:

```
40 A=PEEK(106) -8:POKE 54279,A
```

Die Startadresse ergibt sich dann zwangsläufig:

50 PMSTART=A*256
(eine Seite ist 256 Bytes lang, also Anzahl der freien Seiten. mal 256 ergibt Startadresse in Dezimal).

Auflösung, Farbe und Startposition

Die Hauptarbeit ist nun geschafft, und wenn bisher alles verstanden wurde, werden Sie auch im Folgenden keine Schwierigkeiten mit der PM-Graphik haben. Jetzt sind noch einige Angaben hinsichtlich der Auflösung (entweder ein- oder zweizeilig), Farbe und Startposition des Players zu machen.

Zuerst die Auflösung: Wir wählen eine zweizeilige Auflösung - zwei Fernsehzeilen pro Playerpunkt - und teilen das dem Computer mit, indem wir in Adresse 559 eine 46 hineinschreiben (Wert 62 für einzeilige Auflösung):
60 POKE 559,46
Unser Player soll weiß werden. Wir suchen uns die Farbnummer aus dem Atari-Basic-Manual (oder Anhang A), mulitplizieren Sie mit 16 und addieren dazu den Helligkeitswert. In einer Formel ausgedrückt : Farbwert = Farbnummer * 16 + Helligkeitswert (0-14). Für unser Beispiel ergäbe das dann 0*16+12=12. Dieser Wert wird nun in Adresse 704 geschrieben:
70 POKE 704,12
Die horizontale Ausgangsposition darf zwischen 48 für den linken- und 207 für den rechten Bildschirmrand liegen. Sie wird in Adresse 53248 vermerkt:
80 X=122:POKE 53248, X:REM Horizontale Startposition Player (0). Die vertikale Ausgangsposition speichern wir in der Variablen Y. Sie darf bei einzeiliger Auflösung zwischen acht und 247 und bei zweizeiliger Auflösung im Bereich vier bis 123 liegen:
90 Y=65 :REM Vertikale Auflösung
Jetzt schicken wir noch unsere Player-Daten in eine DATA-Zeile:
200 DATA 24,60,24
Nun noch die alten Daten aus dem Speicher löschen (indem in jede Speicherstelle eine Null geschrieben wird)...
100 FOR L=PMSTART+512 TO PMSTART+640:POKE L,0: NEXT L ...
und die Player Daten hineinschreiben
110 FOR S=PMSTART+512+Y TO PMSTART +512+DATENANZAHL+Y:READ A:POKE S,A:NEXT S
Wir haben drei Player-Daten, die Datenanzahl ist also drei. Zu Beginn des Programms löschen wir auch noch den Bildschirm und schalten auf schwarz/weiß um:
10 GR.0:SETCOLOR 2,0,0
20 Datenanzahl = 3
Bevor der Player dargestellt werden kann, müssen wir die PM-Graphik aktivieren. Dies geschieht in der Adresse 53277:
30 POKE 53277,3
Zur besseren Übersicht das Programm nun noch einmal vollständig als Listing:

```
10  GRAPHICS 0:SETCOLOR 2,0,0:REM
Bildschirm frei und schwarz/weiß
gestellt.
20  DATENANZAHL = 3
30 POKE 53277,3 :REM PM-GRAPHIK akti
viert.
40  A:PEEK  (106) -8:POKE 54279,A:
REM Reservierung des Speicher-
platzes für die PM-Grafik
50 PMSTART=A*256:REM Ermittlung der
Startadresse.
60 POKE 559,46 :REM Zweizeilige
Auflösung
70 POKE 704,12 :REM Farbe weiß für
Player 0
80 POKE 53248,122 :REM Horizontale
Position Player 0
90 Y=65:REM Vertikale Position
Player 0
100  FOR L=PMSTART+512 TO PMSTART+
640:POKE L,0:NEXT L:REM  Player-
streifen löschen.
110  FOR S=PMSTART+512+Y TO PMSTART
+511+DATENANZAHL+Y:READ  A:POKE
S,A:NEXT S:REM Neue Playerdaten in
Steifen eintragen.
120 STOP
200 DATA 24,60,24, :REM Player Daten
```

Fertig zum Start

Wenn wir dieses Programm nun starten, sehen wir in der Mitte des Bildschirms unseren Player. Doch wie bringen wir ihn dazu, sich zu bewegen? Es gibt zwei Bewegungsrichtungen. Die horizontale und die vertikale. Für eine horizontale Bewegung genügt es, den Wert im

horizontalen Positionsregister (Zeile 80) zu ändern. Im Direktmodus geben wir nun einmal folgendes ein: POKE 53248,123 RETURN
Wir beobachten, wie sich der Player ein kleines Stück nach rechts bewegt. Genauer gesagt, das gesamte Playerband hat sich verschoben. Wir halten fest: Durch Veränderung der Werte im horizontalen Positionsregister bewegt sich das gesamte Playerband in der Horizontalen. Die Werte dürfen dabei zwischen 48 für den linken Bildschirmrand und 207 für den rechten liegen. Wir fügen also folgende Zeilen in unser Programm ein:

```
120 ST=STICK(0)
130 IF ST=11 THEN X=X-1:POKE 53248,
X:REM Links Bewegung, 48 -- 207
140 IF ST=7 THEN X=X+1:POKE 53248,
X:REM Rechts Bewegung,48 -- 207
```

Die vertikale Bewegung läßt sich nicht durch Verschieben des Playerbandes bewirken, denn es verläuft ja selbst in der Vertikalen. Man muß also die Daten des Players im Band selbst versetzen, um eine Bewegung zu erreichen. Das könnte man zum Beispiel durch folgende Ergänzung in unserem Programm erreichen:

```
150 IF ST=13 THEN FOR L=DATENANZAHL
TO 0 STEP -1:REM Alte Daten im
Playerband löschen, wenn Stick sich
nach unten (Wert 13) bewegt...
155 IF ST=13 THEN POKE PMSTART
+512+Y+L, PEEK (PMSTART+511+Y+L)
:NEXT L:REM...und Neue um ein Bit
versetzt nach oben eintragen.
159 IF ST=13 THEN Y=Y+1
```

Für die Aufwärtsbewegung gilt dasselbe. Hierbei wird praktisch nur die Schrittrichtung geändert, also nicht STEP-1 sondern STEP +1 und "Y", die Angabe der Zählrichtung, nicht mehr positiv, sondern negativ.

```
160 IF ST=14 THEN FOR L=0 TO DATEN-
ANZAHL:POKE PMSTART +511+Y+L,PEEK
(PMSTART+512+Y+L):NEXT L:Y=Y-1
190 GOTO 120
```

Wenn wir nun dieses Programm starten, werden wir sehen, wie schön unser Player nun das Laufen gelernt hat. Zur Besseren Übersicht das Programm noch einmals als Listing:

<u>Beispiel zum Eintippen:</u>

```
10 GRAPHICS 0:SETCOLOR 2,0,0:REM
BILDSCHIRM FREI UND SCHWARZ/WEIß
GESTELLT.
20 DATENANZAHL=3:REM 3 PLAYER DATEN
30 POKE 53277,3:REM PM-GRAPHIK AKTI-
VIERT
40 A=PEEK(106)-8:POKE 54279,A:REM
RESERVIERUNG DES SPEICHERPLATZES FÜR
DIE PM-GRAPHIK.
50 PMSTART=A*256:REM ERMITTLUNG DER
STARTADRESSE
60 POKE 559,46:REM ZWEIZEILIGE AUF-
LÖSUNG
70 POKE 704,12:REM FARBE PLAYER (0)
-- WEIß
80 X=122:POKE 53248,X:REM HORIZON-
TALE STARTPOSITION PLAYER (0)
90 Y=65:REM VERTIKALE STARTPOSITION
PLAYER (0)
100 FOR L=PMSTART+512 TO PMSTART+
640:POKE L,0:NEXT L:REM PLAYERSTREI-
FEN LÖSCHEN
110 FOR S=PMSTART+512+Y TO PMSTART+
511+DATENANZAHL+Y:READ A:POKE S,A:
NEXT S:REM NEUE PLAYERDATEN IN
STREIFEN EINTRAGEN
120 ST=STICK (0)
130 IF ST=11 THEN X=X-1:POKE
53248,X:REM LINKS BEWEGUNG,48 --
207
140 IF ST=7 THEN X=X+1:POKE
53248,X:REM RECHTS BEWEGUNG,48 --
207
150 IF ST=13 THEN FOR L=DATENANZAHL
TO 0 STEP -1
155 IF ST=13 THEN POKE PMSTART+512+
Y+L,PEEK(PMSTART+511+Y+L) :NEXT
L:REM ALTE DATEN LÖSCHEN UND NEUE
EINTRAGEN.
159 IF ST=13 THEN Y=Y+1:REM RICH-
TUNGSZÄHLER POSITIV, ALSO IM BAND
NACH UNTEN
160 IF ST=14 THEN FOR L=0 TO DATEN-
ANZAHL:POKE PMSTART+511+Y+L,PEEK
(PMSTART+512+Y+L):NEXT L:Y=Y-1:REM
AUFWÄRTSBEWEGUNG
190 GOTO 120
200 DATA 24,60,24
```

<u>Wie jedoch kann ich nun einen zweiten Player auf den Bildschirm bringen?</u>

Ganz einfach: Jeder Player hat drei Dinge, ein eigenes

- O Farbregister (Zeile 70), sein eigenes

o horizontales Positionsregister (Zeile 80,130,140) und seine eigene
o vertikale Position (Zeile 150, 155,160), die sich aus dem eigenen Speicherbereich für die Player Daten (Zeile 100, 110) ergibt.

Man muß also nur diese Adressen austauschen, und schon ist Player Nummer zwei zum Leben erweckt. Nachfolgend nun eine Aufstellung der verschiedenen Adressen für alle vier Player (0-3). Der Vollständigkeit halber auch mit den verschiedenen Kollisionsregistern, aus denen man für jeden einzelnen Player eine Berührung mit einem anderen Player, Missile oder Graphik ermitteln kann.

	PLAYER(0)	PLAYER(1)	PLAYER(2)	PLAYER(3)
FARBREGISTER	704	705	706	707
HORIZONTALES POSITIONSREG.	53248	53249	53350	53251
SPEICHERBEREICH ZWEIZEILIGE AUFLÖSUNG	512-639	640-767	768-895	896-1023
SPEICHERBEREICH EINZEILIGE AUFLÖSUNG	1024-1279	1280-1535	1536-1791	1792-2048
KOLLISIONS-REGISTER	53160	53261	53262	53263

Versuchen Sie nun einmal anhand der Adressen unser "Beipiel zum Eintippen" für Player Nummer eins (zweiter Player) umzuschreiben. Als kleine Hilfe: Die Änderungen müssen in den Zeilen 70,80, 100, 110, 130, 140, 155 und 160 erfolgen. Wenn Sie alles richtig gemacht haben, müßte Ihr Programm dann so aussehen:

```
10  GRAPHICS 0:SETCOLOR 2,0,0:REM BILDSCHIRM FREI UND SCHWARZ/WEIß
GESTELLT.
20 DATENANZAHL=3:REM 3 PLAYER DATEN
30 POKE 53277,3:REM PM-GRAPHIK AKTIVIERT
40 A=PEEK(106)-8:POKE 54279,A:REM RESERVIERUNG DES SPEICHERPLATZES
FÜR DIE PM-GRAPHIK
50 PMSTART=A*256:REM ERMITTLUNG DER STARTADRESSE
60 POKE 559,46:REM ZWEIZEILIGE AUFLÖSUNG
70 POKE 705,12:REM FARBE PLAYER (1) -  WEIß
80 X=122:POKE 53249,X:REM HORIZONTALE STARTPOSITION PLAYER (1)
90 Y=65:REM VERTIKALE STARTPOSITION
PLAYER (1)
100  FOR L=PMSTART+640 TO PMSTART+768:POKE L,0:NEXT L:REM  PLAYER-
STREIFEN LÖSCHEN
110  FOR S=PMSTART+640+Y TO PMSTART+639+DATENANZAHL+Y:READ  A:POKE
S,A:NEXT S:REM NEUE PLAYERDATEN IN STREIFEN EINTRAGEN
120 ST=STICK(0)
```

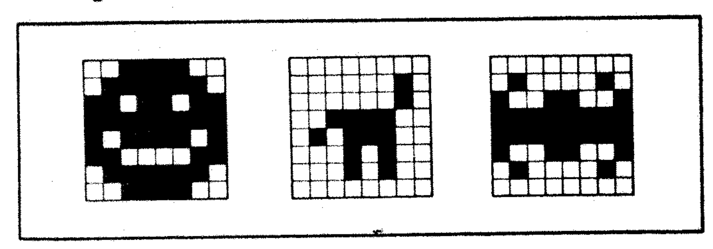
```
130 IF ST=11 THEN X=X-1:POKE 53249,X:REM LINKS BEWEGUNG,48 --- 207
140 IF ST=7 THEN X=X+1:POKE 53249,X:REM RECHTS BEWEGUNG,48 --- 207
150 IF ST=13 THEN FOR L=DATENANZAHL TO 0 STEP -1
155 IF ST=13 THEN POKE PMSTART+640+Y+L, PEEK(PMSTART+639+Y+L):NEXT
L:REM ALTE DATEN LÖSCHEN UND NEUE EINTRAGEN
159 IF ST=13 THEN Y=Y+1:REM RICHTUNGSZÄHLER POSITIV, ALSO IM BAND
NACH UNTEN
160 IF ST=14 THEN FOR L=0 TO DATENANZAHL:POKE PMSTART+639+Y+L,
PEEK(PMSTART+640+Y+L):NEXT L:Y=Y-1:REM AUFWÄRTSBEWEGUNG
190 GOTO 120
200 DATA 24,60,24
```

Wenn Sie diese kleine Übung bestanden haben, können wir Ihnen nun die Hand schütteln und Sie zu eigenen Player-Programmen entlassen, denn nichts ist so wertvoll wie Programmiererfahrung. Erst durch die Erstellung eigener Programme kann man die nötige Routine und insbesondere die Aufspührung von Fehlern im Programm und Ihre Ursache erlernen. Zu den Playern bleibt noch zu sagen, daß Ihre Anwendung besonders in Verbindung mit Graphik unendliche Möglichkeiten bietet.
Wenn man Player jedoch zusammen mit Graphik anwendet, muß man vorher festlegen, was auf dem Bildschirm Priorität haben soll. Mit anderen Worten, soll das Raumschiff oder UFO, hinter der Graphikwand verschwinden oder soll es sichtbar bleiben. Vielfach ist es eine Frage des Geschmacks und man sollte dabei etwas experimentieren, insbesondere mit den Playerfarben.
Folgende Möglichkeiten der Manipulation sind bei dem Prioritätsregister gegeben:
POKE 623,1 : Alle Player vor normaler Graphik.
POKE 623,4 : Normale Graphik vor allen Playern.
POKE 623,2 : Player 0 und 1 vor normaler Graphik, Player 2 und 3 hinter beiden.
POKE 623,8 : Color 0 und Color 1 vor allen Playern. Color 2 und 3 dahinter.

Teil 2: MISSILES
Missiles sind zwei Bit große Objekte, die vorzugsweise als Geschoße Verwendung finden. Es gibt davon insgesamt vier und alle können sich unabhängig von einander bewegen. Sie teilen sich ein Farbregister. Das bedeutet: Sie können nur eine einheitliche Farbe annehmen.
Faßt man Missiles zu einem Streifen zusammen, so erhält man - oh Wunder - einen neuen, vollwertigen Player. Erinnern wir uns. Ein Player ist acht, ein Missile zwei Bit breit. Es liegt also nahe, den fünften Player durch Zusammenfassen der vier Missiles zu bilden. Man sollte dabei jedoch nur im Notfall von den Missiles Gebrauch machen.
Ein überiggebliebener Player läßt sich einfacher in Szene setzen und dürfte auch als Geschoß einem der Missiles vorzuziehen sein.

Ulrich Schmitz

Beispiele für einen Player



Zur Programmierung
Farbregister Missile 0-3: Adresse 707

Speicherbereich:	einzeilig	zweizeilig
Missile 0-3 -->	896-1023	384-511
oder:	768-895	

Horizontale Position Missile 0-3: Adresse 53252-53255
Missile Adresse bei einzeiliger Auflösung: →53252-53255

Missile Adresse bei zweizeiliger Auflösung: →53260-53263
Nachfolgend sind nun die erwähnten Veränderungen in unser Standard-Programm eingebaut:

```
10 GRAPHICS 0:SETCOLOR 2,0,0:REM BILDSCHIRM FREI UND SCHWARZ WEISS
GESTELLT
20 DATENANZAHL=2:REM 2 MISSILE DATEN
30 POKE 53277,3:REM PLAYER MISSILE GRAFIK AKTIVIERT
40 A=PEEK(106)-8:POKE 54279,A:REM RESERVIERUNG DES SPEICHERPLATZES
FÜR DIE PM-GRAFIK
50 PMSTART=A*256:REM ERMITTLUNG DER STARTADRESSE
60 POKE 559,46:REM ZWEIZEILIGE AUFLÖSUNG
70 POKE 704,12:REM MISSILE 0 FARBE WEISS
80 X=122:POKE 53252,X:REM HORIZONTALE STARTPOSITION MISSILE 0
90 Y=100:REM VERTIKALE STRTPOSITION MISSILE 0
100 FOR L=PMSTART+384 TO PMSTART+511:POKE L,0:NEXT L:REM
MISSILESTREIFEN LÖSCHEN
110 FOR S=PMSTART+384+Y TO PMSTART+383+DATENANZAHL+Y:READ A:POKE
S,A:NEXT S:REM NEUE MISSILE DATEN IN STREIFEN EINTRAGEN
120 ST=STICK(0)
130 IF ST=11 THENX=X-1:POKE 53252,X:REM HORIZONTALE POSITION MISSILE 0
140 IF ST=7 THEN X=X+1:POKE 53252,X:REM HORIZONTALE POSITION MISSILE 0
150 IF ST=13 THEN FOR L=DATENANZAHL TO 0 STEP -1
155 IF ST=13 THEN POKE PMSTART+384+Y+L,PEEK(PMSTART+384+Y+L):NEXT
L:REM ALTE DATEN LÖSCHEN UND NEUE EINTRAGEN
159 IF ST=13 THEN Y=Y+1:REM RICHTUNGSZÄHLER POSITIV, ALSO IM BAND NACH
UNTEN
160 IF ST=14 THEN FOR L=0 TO DATENANZAHL:POKE
PMSTART+383+Y+L,PEEK(PMSTART+384+Y+L):NEXT L:Y=Y-1:REM
AUFWÄRTSBEWEGUNG
190 GOTO 120
200 DATA 11,11
```

Hilfe zum Erstellen eigener Programme

Funktionen auf dem Bildschirm	Erzeugung	Folge
↑	ESC/CTRL/-	Cursor nach oben
[E] ↑	ESC/ESC/ESC/CTRL/-	Bildschirmdarstellung von ↑
↓	ESC/CTRL/=	Cursor nach unten
[E] ↓	ESC/ESC/ESC/CTRL/=	Bildschirmdarstellung von ↓
→	ESC/CTRL/*	Corsor nach rechts
[E] →	ESC/ESC/ESC/CTRL/*	Bildschirmdarstellung von →
←	ESC/CTRL/*	Cursor nach links

ESC/ESC/ESC/CTRL/*	Bildschirmdarstellung von
ESC/CTRL/	Bildschirm löschen
ESC/ESC/ESC/CTRL/	Bildschirmdarstellung von
ESC/BACK	Cursor nach links, das dort befindliche Zeichen wird gelöscht
ESC/ESC/ESC/BACK	Bildschirmdarstellung von
ESC/TAB	Cursor an den nächsten Tabulator-Stopp bewegen
ESC/ESC/ESC/TAB	Bildschirmdarstellung von
ESC/SHIFT/BACK	Zeile, in der sich der Cursor befindet, wird gelöscht
ESC/ESC/ESC/SHIFT/BACK	Bildschirmdarstellung von
ESC/SHIFT/	Einfügen einer Zeile unterhalb des Cursors
ESC/ESC/ESC/SHIFT/	Bildschirmdarstellung von
ESC/CTRL/TAB	Tabulator-Stopp wird dort, wo der Cursor steht gelöscht
ESC/ESC/ESCF/CTRL/TAB	Bildschirmdarstellung von
ESC/SHIFT/TAB	Tabulator-Stopp wird an Cursor-Position eingefügt
ESC/ESC/ESC/SHIFT/TAB	Bildschirmdarstellung von
ESC/CTRL/2	Lautsprecher des Computers ertönt
ESC/ESC/ESC/CTRL/2	Bildschirmdarstellung von
( ) ESC/CTRL/BACK	Das rechts vom Cursor stehende Zeichen wird gelöscht, Rest der Zeile rückt nach.

	ESC/ESC/ESC ( ) CTRL/ BACK	Bildschirmdarstellung von
	( ) ESC/CTRL/	Rechts vom Cursor wird ein Leerzeichen eingefügt. Rest der Zeile wird nach rechts verschoben.
	ESC/ESC/ESC ( ) CTRL/	Bildschirmdarstellung von

PRAXISBEISPIEL:

```
10 Print "     Peter"
RUN
```

Dieses Kurzprogramm erzeugt zuerst eine Bildschirmlöschung, dann schreibt der Computer "Peter".

Dasselbe läßt sich auch durch den ATASCII-Code erreichen.
Beispiel:

```
10 Print CHR$(125);"Peter"
RUN
```

Diese Umwandlung in einen Code (CHR$) läßt sich mit jedem Zeichen und jeder Funktion nachvollziehen.

Große PEEK- und POKE-Liste

Adresse (Hex)	(Dez.)	Bez.	Bemerkungen
2,3	2,3	CASINI	Wenn Kassettenboot erfolgreich abgeschlossen wurde, JSR zu vorheriger Adresse.
4	4	RAMLO	RAM-Zeiger für den Vektortest
6	6	TRAMSZ	Zeitweises Register für die Speicher Gr.
7	7	TSTDAT	RAM-Test Register für Daten
8	8	WARMST	0 = Einschalten des Gerätes
			− 1 = System-Reset gedrückt
			255 = Normal Wert
			Programme bleiben bei Sprung zu dieser Adresse erhalten
			Boot Kennung:
9	9	BOOT	Bit 1 gesetzt: auf RESET Sprung zu DOSINI
			Bit 2 gesetzt: auf RESET Sprung zu CASINI
			Sind Bit 1 und 2 gesetzt, erfolgt zuerst Cassetteninitialisierung
			Start-Zeiger für Diskettenprogramme
A, B	10, 11	DOSVEC	Wird zu Speicherung der Adresse verw., bei der nach ladeb des DOS das
C, D	12, 13	DOSINI	Benutzer-Programm startet. Mit einem indirekten JSR hierher läßt es sich initialisieren.
			Enthält höchste nutzbare Benutzer-Adresse
E, F	14, 15	APPMHI	Pokey Interrupt Zeiger
10	16	POKMSK	Bit 7 = Interrupt für Break Taste
			Bit 6 = Interrupt für andere Tasten
			Bit 5 = Interrupt für serielle Dateneing.
			Bit 4 = Interrupt für serielle Datenausg.
			Bit 3 = Interrupt für Datenausgabe Ende K.

Adresse (Hex)	(Dez.)	Bez.	Bemerkungen
			Bit 2 = Interrupt Timer 4 möglich Bit 1 = Interrupt Timer 1 möglich POKE 16,64 zusammen mit POKE 53774,64 blockiert beispielsweise die BREAK-Taste
11	17	BRKKEY	Abfrage der Break-Taste. Normal = − 1 wird bei Tastaturanwendung auf 0 gesetzt
622		LINBUF	POKE 622,255 schaltet automatisches Fine Scrolling ein.
756		CHBAS	POKE 756,204 aktiviert den integr. intern. Zeichensatz.
53769		KBCODE	Für die Help-Taste bekommt man folgende Werte
	17	Help-Taste gedrückt.	
	81		Help- und Shift-Taste gedrückt.
	145		CTRL und Help-Taste gedrückt.

Es ist dabei darauf zu achten, daß vor jeder neuen Abfrage 255 in die Adresse gepoked wird.

Adressen im Dezimal- und Hexadezimal-System

Adresse in
(Bezeichnung) **(Dez.)** **(Hex.)**

★★★ RTCLOK 18,19,20 12–14 ★★★
Interne Uhr... Alle 1/50 Sekunde wird der Inhalt von Adresse 20 um eins erhöht, bis 255 erreicht wird. Dann wird 20 auf 0 gesetzt und dafür 19 um 1 incrementiert. Dies wird gemacht, bis der Inhalt von Speicherstelle 19 auf 255 steht, dann werden 19 und 20 auf Null gesetzt und 18 entsprechend incrementiert. Dies kann man mit Hilfe eines Programms leicht ausprobieren:

```
100 SEKUNDEN =
   INT((PEEK(18)★65536+PEEK(19)★256+PEEK(20))/60)
```

★★★ BUFADR 21,22 15,16 ★★★
Indirektes Buffer-Adressen-Register. Wird als vorrübergehender PAGE 0 Pointer auf den momentanen Disk Buffer verwendet.

★★★ ICCOMT 23 17 ★★★
>>>>Command for Vektor ??????

★★★ DSKFMS 24,25 18,19 ★★★
>>>>Disk file manager pointer ?????

★★★ DSKUTL 26,27 1A,1B ★★★
>>>>Disk Utilities Pointer ?????

★★★ PTIMOT 28 1C ★★★
Drucker-Sperrzeit (TIMEOUT). Eine für den 825 typische Sperrzeit sind 5 Sekunden. Initialisiert auf 30 Sekunden.

★★★ PBPNT 29 1D ★★★
Zeiger auf Printer Buffer. Läuft im Drucker Buffer von 0 bis zur in PBUFSZ festgelegten Höchstzahl.

★★★ PBUFSZ 30 1E ★★★
Länge des Drucker Buffers im jeweiligen Mode.

Normal	= 40 Byte
Doppelt breit	= 20 Byte
Seitwärts	= 29 Byte
Status	= 4 Byte

★★★ PTEMP 31 1F ★★★
Der Drucker Handler verwendet dieses vorrübergehende Register, um den Wert des Zeichens zu speichern, das auf den Drucker ausgegeben werden soll.

★★★ IOCBAS 32 20 ★★★
★★★ ZIOCB 32 20 ★★★
★★★ ICHIDZ 32 20 ★★★
'Handler Index Nummer'. Wird vom OS als ein Index auf die Filenamentabelle für eine geöffnete File gesetzt. Wird auf 255 ($FF) gesetzt, wenn keine File geöffnet ist.

★★★ ICDNOZ 33 21 ★★★
File Laufwerk Nummer. Wird vom OS auf 1 bis 4 gesetzt, für das zu benutzende Laufwerk.

★★★ ICCOMZ 34 22 ★★★
Befehlsbyte. Aus dem Benutzerprogramm. Legt fest, wie der Rest der IOCBs formatiert ist.

★★★ ICSTAZ 35 23 ★★★
Statusbyte, das von dem Device erwidert wird.

★★★ ICBALZ/H 36,37 24,25 ★★★
Bufferadresse für den Datentransfer oder Adresse des Filenamens für Befehle wie OPEN, STATUS, etc.

★★★ ICBLLZ/HZ 40,41 28,29 ★★★
Bufferlängenzähler für PUT/GET Befehle. Wird bei jedem übertragenen Byte um 1 erniedrigt.

★★★ ICAX1Z 42 2A ★★★
Hilfsinformation, die beim Öffnen benutzt wird, um die Art des Dateizugriffs festzulegen.

★★★ ICAX2Z 43 2B ★★★
Einige Funktionen des Seriellen Ports benutzen dieses Byte.

★★★ ICAX3&4Z 44,45 2C,2D ★★★
Die Stelle, um die Disksektornummer von NOTE und POINT zu übertragen.

★★★ ICAX5Z 46 2E ★★★
Das Byte in obigem Sektor.

★★★ ICAX6Z 47 2F ★★★
Noch nicht belegt!

★★★★★★★★★★★★★★★★★

★★★ SOUNDR 65 41 ★★★
Eingabe/Ausgabe-Laut-Flag... Versuche 0, um den Computer während einiger Funktionen Disketten oder Cassetten Einlesen nicht 'biepen' zu lassen.

★★★ ?????? 66 42 ★★★
1 = Repeat Funktion der Tasten ausschalten und ändere den Ton des CTRL/2 Summers.

★★★ ATRACT 77 4D ★★★
Ein Wert kleiner als 128 ist der normale Wert. 128 ergibt eine kleinere Helligkeit und wechselt die Farben um den Bildschirm zu schonen. Wenn ein Programm die Tastatur nicht benötigt, ist es gut, alle 5–9 Minuten eine Zahl<128 in diese Speicherzelle zu schreiben. Dies ist besonders bei Spielen mit Joysticks wichtig.

★★★ DRKMSK 78 4E ★★★
254 = Normale Helligkeit

★★★ LMRGIN 82 52 ★★★
Linker Bildschirmrand. Ist normalerweise mit 2 belegt.

★★★ RMRGIN 83 53 ★★★
Rechter Bildschirmrand. Ist normal mit 39 belegt. Diese beiden Adressen können benutzt werden, um den Bildschirm an einer beliebigen Stelle anfangen und aufhören zu lassen.

★★★ ROWCRS 84 54 ★★★
Reihe auf dem Bildschirm, in dem sich der Grafikcursor gerade befindet.

★★★ COLCRS 85,86 55,56 ★★★
Spalte auf dem Bildschirm, in dem sich der Cursor gerade befindet.
Diese zwei Adressen können benutzt werden, um den Cursor an eine bestimmte Stelle zu setzen oder seine Position auf dem Bildschirm festzustellen.

★★★ CRMODE 87 57 ★★★
Wird benutzt, um das OS. auszutricksen, daß es glaubt, ein anderer Grafikmodus als der ursprüngliche ist eingeschaltet. Diese Adresse ist auch beim Erstellen von Display Lists wichtig, um PLOT und PRINT Befehle zu ermöglichen. Wird mit Werten von 0 bis 8 (wie die normalen Grafikstufen) belegt. Nähere Informationen sind im DISPLAY LIST TUTORIAL von Santa Cruz Software (in Deutschland Münzenloher GmbH, Holzkirchen) gegeben.

★★★ SAVMSC 88,89 58,59 ★★★
Niedrigste Adresse des Bildschirmspeichers. Die Daten in dieser Adresse werden in die obere linke Ecke geschrieben und die nächste Anzahl von Bytes wird darauf folgend geschrieben.

★★★ OLDROW 90 5A ★★★
Momentane Reihe des Grafikcursors

★★★ OLDCOL 91,92 5B,5C ★★★
Momentane Spalte des Grafikcursors

★★★ OLDCHR 93 5D ★★★
Daten unter dem Cursor

★★★ NEWROW 96 60 ★★★
Punkt (Reihe), zu der DRAWTO gehen soll

★★★ NEWCOL 97,98 61,62 ★★★
Punkt (Spalte), zu der DRAWTO gehen soll

★★★ RAMTOP 106 6A ★★★
Höchste Adresse des RAM Bereichs. Angegeben als Anzahl der Pages (Ein Page = 256Bytes im RAM). Dies wird oft dazu benutzt, den Computer glauben zu machen, daß er einen kleineren Speicher als vorhanden hat, und so einen geschützten Bereich für Maschinenroutinen zu schaffen. Dazu wird der Befehl gegeben POKE 106,PEEK(106) – # der Pages, die geschützt werden sollen.

★★★ LOMEM 128,129 80,81 ★★★
Token Ausgabe Buffer, der Buffer, den ABASIC benutzt, um eine Befehlszeile in Tokens zu übersetzen. Dieser Buffer liegt am Ende des zur Verfügung stehenden RAMs.

★★★ VNTP 130,131 82,83 ★★★
Liste der Variablennamen; eine Liste aller Variablennamen, die im Programm verwendet wurden. Sie werden als ATASCII-Zeichen gespeichert. Die Namen werden in der Reihenfolge gespeichert, in der sie eingegeben werden. Die Variablen werden dabei in drei Gruppen eingeteilt:
(1) Numerische Variablen – Beim letzten Buchstaben des Namens ist das MOST SIGNIFICANT BIT (MSB) gesetzt.
(2) Textvariablen – Das letzte Zeichen ist ein '$', bei dem das MSB gesetzt ist.
(3) Matrix Variablen – Das letzte Zeichen ist ein '(', dessen MSB gesetzt ist.

★★★ VNTD 132,133 84,85 ★★★
Ende der Variablentabelle. Dieser Zeiger zeigt, wenn weniger als 128 Variablen verwendet werden, auf ein Dummybyte, ansonsten auf das letzte Byte des letzten Namens. Die Variablenliste ist also auf jeden Fall 128 Bytes lang!

★★★ VVTP 134,135 86,87 ★★★
Variablen Wert Tabelle; Diese Tabelle enthält die aktuelle Information für jede Variable. Dabei sind für jede Variable in der Variablentabelle 8 Bytes reserviert.

★★★ STMTAB 136,137 88,89 ★★★
Befehlstabelle; diese Tabelle enthält allen eingegebenen und in TOKENs verwandelten Code.

★★★ STMCUR 138,139 8A,8B ★★★
Momentaner Befehl. Dieser Zeiger wird von ABASIC benutzt, um einen bestimmten Befehl festzulegen.

★★★ STARP 140,141 8C,8D ★★★
Speicherbereich für Text und Felder; Dieser Block enthält alle Text- und Felddaten. Zeichen werden dabei in ATASCII gespeichert, d. h. jedes Zeichen belegt ein Byte. Arrays benötigen 6 Bytes für eine Zahl. Die Größe dieses Feldes wird durch das DIM Statement festgelegt.

★★★ RUNSTK 142,143 8E,8F ★★★
Run Time Stack; dieser Software Stack enthält die GOSUB und FOR/NEXT Ursprünge.

★★★ MEMTOP 144,145 90,91 ★★★
Von der hier festgelegten Adresse bis zu dem Anfang der Displaylist kann das ABASIC Programm gehen. Die FRE-Funktion gibt den Wert von (HIMEM (2E5,2E6) minus (MEMTOP) an.

Vorsicht: ABASIC MEMTOP ist nicht identisch mit der im OS MEMTOP genannten Variablen.

★★★ STOPLN 186,187 BA,BB ★★★
Zeilennummer, in der ein STOP oder TRAP aufgetreten ist.

★★★ ERRSAVE 195 C3 ★★★
Fehlernummer, die den STOP oder TRAP ausgelöst hat.

★★★ PTABW 201 C9 ★★★
Schrittweite für TAB. Ist normal auf 10 gesetzt. Wird benützt um die Schrittweite beim TAB zu ändern.

★★★ FRO 212–217 D4-DB ★★★
Fließkommaregister 0. (Der Wert, der von der USR Funktion zurückgegeben wird.)

★★★ FR1 224–229 E0–E5 ★★★
Fließkommaregister 1

★★★ CIX 242 F2 ★★★
Zeichenindex

★★★ INBUFF 243 F3 ★★★
Zeiger zum Texteingabebuffer

★★★ RADFLG& 251 FB ★★★
RAD/DEG Schalter, (RAD=0, DEG=6)

★★★ VDSLST 512,513 200,201★★★
Display List Interrupt Vektor (DLI). Man muß zuerst POKE 54268 vor ein DLI setzen. DLIs werden benützt, um den normalen Fluß der Ausführung der Display List anzuhalten und für einige Mikrosekunden etwas anderes zu tun. So ist es zum Beispiel möglich, Musik unabhängig von der Programmausführung taktgerecht zu spielen.

★★★ VPRCED 514,515 202,203★★★
Serieller Proceed

★★★ VINTER 516,517 204,205★★★
Serieller Interrupt

★★★ VBREAK 518,519 206,207★★★
Vektor zur BREAK Anweisung

★★★ VKEYBD 520,521 208,209★★★
Vektor zum Tastatur Interrupt

★★★ VSERIN 522,523 20A,20B★★★
Signal für die Bereitschaft, den seriellen Bus zu benutzen

★★★ VSEROR 524,525 20C,20D★★★
Signal für die Bereitschaft, serielle Daten zu übermitteln

★★★ VSEROC 526,527 20E,20F★★★
Signal, daß die Übertragung serieller Daten beendet ist

★★★ VTIMR1 528,529 210,211★★★
POKEY Timer Zeiger

★★★ VTIMR2 530,531 212,213★★★
POKEY Timer 2 Zeiger

★★★ VTIMR3 532,533 214,215★★★
POKEY Timer 3 Zeiger

★★★VIMIRQ 534,535 216,217★★★
IRQ immidiate Zeiger (normalerweise)

★★★★★★★★★★★★★★★★★

SYSTEM TIMERS

Diese Timer zählen jede 1/50 Sekunde um eins herunter, bis sie 0 erreicht haben. Man kann sie für Zeitmessungen, unabhängig von der Programmausführung verwenden.

★★★ CDTMV1 536,537 218,219★★★
Wert für den SYSTEMTIMER 1

★★★ CDTMV2–5 538–545 21A–221F★★★
Wert für den SYSTEMTIMER 2–5

★★★ VVBLKI 546,547 222,223★★★
Vertical Blank Immediate

★★★ VVBLKD 548,549 224,225★★★
Vertical Blank Deferred

★★★ CDTMA1 550,551 226,227★★★
Sprungadresse für SYSTEMTIMER 1

★★★ CDTMA2 552,553 228,229★★★
Sprungadresse für SYSTEMTIMER 2

★★★ CDTMF3–5 554–558 22A–22B★★★
Flag für SYSTEMTIMER 3–5

★★★★★★★★★★★★★★★★★

★★★ SDMCTL 559 22F ★★★
Direkt MEMORY access enable (DMA), s. 54272

★★★ SDLSTL 560,561 230,231★★★
Zeiger für Display List. Gibt die Startadresse der Display List an, die verändert werden kann. s. DISPLAY LIST TUTORIAL von SANTA CRUZ SOFTWARE (in Deutschland Münzenloher GmbH, Holzkirchen).

★★★ SSKCTL 562 232 ★★★
Kontrolle für Seriellboard

★★★ LPENH 564 234 ★★★
Horizontale Position des Light Pens

★★★ LPENV 565 235 ★★★
Vertikaler Wert des Light Pen

★★★ COLDST 580 244 ★★★
1 = Coldstart / 0 = Normal
POKE 580,1 und Drücken von SYSTEM RESET entspricht dem AUS- und ANSCHALTEN des Computers (Kaltstart).

★★★ GPIOR 623 26F ★★★
Shadow von 53257

★★★ PADDLO 624 270 ★★★
Wert von Paddle 0, kann von 0 bis 228 gehen

★★★ PADDL1–7 625–631 271–277★★★
Wie PADDLO für die anderen Drehregler

★★★ STICKO–3 632–635 278–27B★★★

Wert, von der der Position des Joysticks bestimmt (s. Reference Manual)

★★★ PTRIGO–7 636–643 27C–283★★★
Knopf von Drehregler:
0 = gedrückt / 1 = nicht gedrückt

★★★ STRIGO–3 644–647 284–287★★★
Knopf von Joystick:
0 = gedrückt / 1 = nicht gedrückt

★★★ TXTROW 656 290 ★★★
Reihe, in der sich der Text Cursor befindet. (0–3) (in Textfenster bei Grafikmode).

★★★ TXTCOL 657,658 291,292★★★
Spalte, in der sich der Text Cursor befindet. (0–39).

★★★ TXTMSC 660,610 294,295★★★
Obere linke Ecke des Textfensters

★★★ TABMAP 675–689 2A3–2B1★★★
Eine 1 in jeder Bit Position von jeder dieser 120 Bytes verursacht einen Tab-Stop.
Um alle Tabs zu löschen, wird in jede Adresse 0 geschrieben.

★★★ INVFLG 694 2B6 ★★★
128 = Invers Video (Atari Taste)
0 = Normal

★★★ SHFLOK 702 2BE ★★★
0 = Kleinschrift
67 = Großschrift

★★★ BOTSCR 703 2BF ★★★
Anzahl der auf TV auszugebenden Bildschirmzeilen (24, 4 oder 0)

★★★ PCOLRO–3 704–707 2C0–2C3★★★
Farbe für Player & Missile 0–3

★★★ COLOR0 708 2C4 ★★★
Farbregister 0 (SETCOLOR); Großbuchstaben

★★★ COLOR1 709 2C5 ★★★
Farbregister 1 (SETCOLOR 1); Kleinbuchstaben

★★★ COLOR2 710 2C6 ★★★
Farbregister 2 (SETCOLOR 2); Inverse Großschrift

★★★ COLOR3 711 2C7 ★★★
Farbregister 3 (SETCOLOR 3); Inverse Kleinschrift

★★★ COLOR4 712 2C8 ★★★
Farbregister 4 (SETCOLOR 4); Grafikbereich

★★★ MEMTOP 741,742 2E5,2E6★★★
Zeiger für höchste Adresse des OS

★★★ MEMLOW 743,744 2E7,2E8★★★
Zeiger für niedrigste Adresse des OS

★★★ DVSTAT 746 2EA ★★★
Kanal (Device) Status

★★★ CRSINH 752 2F0 ★★★
0 = Cursor an
nicht 0 = Cursor aus

★★★ KEYDEL 753 2F1 ★★★
0 = keine Taste gedrückt
3 = eine Taste gedrückt

★★★ CHACT 755 2F3 ★★★
Register für Zeichenmode
0 = Normales Inverse
1 = Alle Inverszeichen verschwunden
2 = Alle Inverszeichen nicht invers
3 = Alle Inverszeichen weiß
4–7 = wie 0–3, aber auf den Kopf gestellt

★★★ CHBAS 756 2F4 ★★★
Register für Zeichenbasis (Character Base)
224 = Großschrift
226 = Kleinschrift

★★★ ATACHR 763 2FB ★★★
Das letzte ATASII Zeichen, das gelesen oder geschrieben worden ist oder der Wert des Grafikpunktes. Hier wird die Farbe für den FILL Befehl abgelegt.

★★★ CH 764 2FC ★★★
Der interne Code für die letzte gedrückte Taste. 255 löscht.

★★★ FILDAT 765 2FD ★★★
Farbe für den FILL Befehl bei Grafik (XIO)

★★★ DSPFLG 766 2FE ★★★
Display Flag:
0 = Normal
1 = Control Zeichen

★★★ SSFLAG 767 2FF ★★★
Start / Stop Flag für PAGING; wird gesetzt mit CTRL/1.

★★★ DDEVIC 768 300 ★★★
Device Bus Identifikation

★★★ DUNIT 769 301 ★★★
Disk Drive Nummer (1–4)

★★★ DCOMND 770 302 ★★★
Auszuführender Diskettenbefehl

★★★ DSTATS 771 303 ★★★
Status Code des Device

★★★ DBUFLO/H 772,773 304,305★★★
Adresse, auf die die Daten eines Diskettensektors geschrieben/gelesen werden.

★★★ DTIMLO 774 306 ★★★
Zeit, die der Handler für eine Operation benutzen darf.

★★★ DBYTLO/H 776,777 308,309★★★
Anzahl von Bytes, die an oder von der Disk übermittelt wurden als Ergebnis von Funktionen.

★★★ DAUX1/2 778,779 30A,30B★★★
Nummer des Diskettensektors, auf den geschrieben / von dem gelesen wird.

★★★ HATABS 794–831 31A–33F★★★

Liste für Handler Adressen (3 Bytes pro Handler). Das Maximum sind 12 Eingänge. Beinhaltet das einzelne Zeichen für den Devicenamen (K,P,E,S,C,D) und die Handleradresse für jeden Eingang. Der Rest der 28 Bytes sind Nullen. Ein Handler ist eine Zusammenstellung von Anweisungen, die dem Computer sagen, wie der den Bildschirm, die Diskette ... benutzen soll.

★★★ IOCBO 832–847 340–34F★★★
Kontrollblock 0 für Input/Output

★★★ ICPTL/H 838,839 346,347★★★
POKE 838,166 und POKE 839,238 bedeutet, daß alles, was normalerweise auf dem Bildschirm ausgegeben wird, nun an den Drucker geht. Die normalen Werte sind: POKE 838,163 und POKE 839,246. Die Benutzung für andere Geräte ist auch möglich.

★★★ ICAX1 842 34A ★★★
13 = Lesen vom Bildschirm
12 = Auf Bildschirm schreiben
Siehe auch TRICKY TUTORIAL I (Display List).

★★★ IOCB1-7 848–959 350–3BF★★★
Input/Output Kontrollblock (IOCB) 1–7

★★★ LBUFF 140B 580 ★★★
Textbuffer

★★★ ?????? 1664/5 680 ★★★
Zeiger, der von der Dumproutine für die Variablennamenliste benutzt wird. Der Pointer ist im Speicher, der weder von OS noch von BASIC benutzt wird, gespeichert.

★★★ ?????? 1802 70A ★★★
Die Anzahl der Diskettenstationen des Systems sind hier gespeichert. Normalerweise auf 2 gesetzt. Wenn mehr oder weniger vorhanden sind, sollte die Anzahl hier geändert werden, und dann das geänderte DOS wieder auf Diskette gespeichert werden (H-Option). Zum Beispiel: Angeschlossene Laufwerke: 1,2,4 => 00001011 (Binär) => 11 (Dezimal).

★★★ ?????? 1913 779 ★★★
80 = Schaltet Vergleichstest beim Schreiben aus
87 = Schaltet wieder an
80 zu benutzen ist wesentlicher schneller, aber meist entstehen dann Fehler! (Benutzen auf eigene Gefahr!)

★★★ VDELAY 53276 D01C ★★★
Vertikaler Delay

★★★ GRACTL 53277 D01D ★★★
??

★★★ HITCLR 53278 D01E ★★★
Jede Zahl ungleich 0 bewirkt das Löschen des Kollisionsregisters.

★★★ CONSOL 53279 D01F ★★★
Abfrage, welche der Zusatztasten gedrückt wurden. 8 löscht das Register. Eine Zahl von 0–7 läßt den Lautsprecher klicken. X bedeutet, daß die entsprechende Taste gedrückt wurde.

Taste	★	1	2	3	4	5	6	7	8	9	0
★★★★★★★★★★★★★★★★★★★★★★★★											
Start	★		X		X			X			X
Select	★	X			X	X					X
Option	★	X	X	X							X
★★★★★★★★★★★★★★★★★★★★★★★★											

Das AUDF# Register entspricht dem Schalter für die Soundkanäle 0–3. Die AUDC# sind die Kontrollregister für die Lautstärke.

★★★ AUDF1 53760 D200 ★★★
Geräuschkanal 1: Frequenz

★★★ AUDC1 53761 D201 ★★★
Geräuschkanal 1: Kontrolle

★★★ AUDF2 53762 D202 ★★★
Geräuschkanal 2: Frequenz

★★★ AUDC2 53763 D203 ★★★
Geräuschkanal 2: Kontrolle

★★★ AUDF3 53764 D204 ★★★
Geräuschkanal 3: Frequenz

★★★ AUDC3 53765 D205 ★★★
Geräuschkanal 3: Kontrolle

★★★ AUDF4 53766 D206 ★★★
Geräuschkanal 4: Frequenz

★★★ AUDC4 53767 D207 ★★★
Geräuschkanal 4: Kontrolle

★★★ AUDCTL 53768 D208 ★★★
Audio Control

Wichtige Memory Adressen

★R★ KBCODE 53769 D209 ★★★
Tastaturcode (764)

★W★ STIMER 53769 D209 ★★★
Start Timer

★R★ RANDOM 53770 D20A ★★★
Zufallszahlen Generator

★W★ SKRESET 53770 D20A ★★★
Setze seriellen Port Status (53775) zurück.

★W★ POTGO 53771 D20B ★★★
Starte die POT SCAN Sequenz

★W★ SEROUT 53773 D20D ★★★
Serielle Port Ausgabe

★R★ SERIN 53773 D20D ★★★
Serielle Port Eingabe

★W★ IRQEN 53774 D20E ★★★
Ermöglichen von Interrupt Request. Wenn hier ausgeschaltet wird, arbeiten die Tasten nicht mehr, die ein Interrupt haben (BREAK, SYSTEM RESET, ...). 0 schaltet aus, 16 ein.

★R★ SKCTL 53775 D20F ★★★
(562) Serielle Port Kontrolle
3 = kein Ton beim INPUT/OUTPUT. Die Adresse beinhaltet 251 bei den meisten gedrückten Tasten, 255, wenn keine Taste gedrückt ist und 247 bei Shift.

★R★ SKSTAT 53775 D20F ★★★
Liest seriellen Port Status.

===> Für 53776–54015 wie 53760–53775 <===

★WR PORTA 54016 D300 ★★★
Liest oder schreibt Daten von/an 'CONTROLLER JACK' 1 und 2 wenn das Bit 2 von PACTL =1. Wenn das Bit 2 =0 werden Daten geschrieben. (632 für JACK 1, 633 für JACK 2)

★WR PORTB 54017 D301 ★★★
Liest oder schreibt Daten von/an 'CONTROLLER JACK' 3 und 4. (wie oben). (634 für JACK 3, 635 für JACK 4)

★W★ PACTL 54018 D302 ★★★
PORT A Kontrolle:
60 Schaltet Kassettenmotor aus
52 Schaltet Kassettenmotor an
Wird benützt, um Musik oder Sprache vom Kassettenrecoder auf den Lautsprecher des Fernsehers zu übertragen.

★W★ PBCTL 54019 D303 ★★★
PORT B Kontrolle

===> 54020–54271 wie 54016–54019 <===

★W★ DMACTL 54272 D400 ★★★
(559) Direkte Kontrolle für Speicherzugriff (DMA). Schaltet DMA ein, gibt ein oder zwei Zeilenauflösung für Players, schaltet PM ein.
Aus den folgenden Möglichkeiten müssen die entsprechenden Werte addiert und dann in 559 geschrieben werden:

Weites Playfield	3ö
Standard Playfield	2ö
Schmales Playfield	1 Nur eins auswählen
Kein Playfield	0/
Ermögiche Missile DMA	4
Ermögliche Player DMA	8
2 Zeilen dicke PLAYER	0
1 Zeilen dicke PLAYER	16
Ermögliche Instruktions DMA	32

Siehe auch PM TUTORIAL von Santa Cruz Software (in Deutschland Münzenloher GmbH, Holzkirchen).

★★★ CHACTL 54273 D401 ★★★
(755) Zeichenkontrolle:
4 = Zeichen auf dem Kopf
2 = Wie Atari Taste
1 = Normale Zeichen

★★★ DLISTL/H 54274/5 D402/3 ★★★
(560/1) Zeiger für Display List. Diese Adresse sagt, wo das OS die Anweisungen, in welchem Mode, und welche Daten auf den Bildschirm geschrieben werden, für den Computer hingelegt hat. Auch hier gibt es ein TUTORIAL.

★★★ HSCROL 54276 D404 ★★★
Horizontales Scrollen wird ermöglicht.
Ausführliche Beschreibung s. TUTORIAL#2.

★★★ VSCROL 54277 D405 ★★★
Vertikales Scrollen ermöglichen.

★★★ PMBASE 54279/80 D407/8 ★★★
Player/Missile Basis Adresse

★★★ CHBASE 54281 D409 ★★★
(756) Zeichensatz Basis Adresse. Man kann damit einen eigenen Zeichensatz programmieren. Siehe Programm von IRIDIS.

★★★ WSYNC 54282 D40A ★★★
Warte auf horizontale Synchronisation.

★★★ VCOUNT 54283 D40B ★★★
Vertikaler Zeilenzähler

★★★ PENH 54284 D40C ★★★
(564) Horizontale Position des Light Pen

★★★ PENV 54285 D40D ★★★
Vertikale Position des Light Pen

★★★ NMIEN 54286 D40E ★★★
192 = erlaubt ein Display List Interrupt

★W★ NMIRES 54287 D40F ★★★
Setze NMIST zurück.

★R★ NMIST 54287 D40F ★★★
NMJ Status
===> 54288–54277 wie 54272–54287 <===

★★★ AFP 55296 D800 ★★★
Umwandlung Ascii –> Floating Point

★★★ FASC 55526 D8E6 ★★★
Umwandlung Floating Point –> Ascii

★★★ IFP 55722 D9AA ★★★
Umwandlung Integer –> Floating Point

★★★ FPI 55760 D9D2 ★★★
Umwandlung FP –> Integer

★★★ ZFRO 55876 DA44 ★★★
Lösche FRO.

★★★ ZF1 55878 DA46 ★★★
Lösche FP Nummer.

★★★ FSUB 55904 DA60 ★★★
Floating Point Subtraktion

★★★ FADD 55910 DA66 ★★★
Floating Point Addition

★★★ FMUL 56027 DADB ★★★
Floating Point Multiplikation

★★★ FDIV 56104 DB28 ★★★
Floating Point Division

★★★ PLYEVL 56640 DD40 ★★★
Floating Point Polynom Entwicklung

★★★ FLDOR 56713 DD89 ★★★
Lade FP Nummer.

★★★ FLDOP 56718 DD8D ★★★
Lade FP Nummer.

★★★ FLD1R 56728 DD98 ★★★
Lade FP Nummer.

★★★ FLD1P 56732 DD9C ★★★
Lade FP Nummer.

★★★ FSTOR 56743 DDA7 ★★★
Speichere FP Nummer.

★★★ FSTOP 56747 DDAB ★★★
Speichere FP Nummer.

★★★ FMOVE 56758 DDB6 ★★★
Verschiebe FP Nummer.

★★★ EXP 56768 DDCO ★★★
Bilde den FP Exponenten zur Basis e.

★★★ EXP10 56780 DDCC ★★★
Bilde den FP Exponenten zur Basis 10.

★★★ LOG 57037 DECD ★★★
Bilde den FP Logarithmus zur Basis e.

★★★ LOG10 57041 DED1 ★★★
Bilde den FP Logarithmus zur Basis 10.

★★★★★★★★★★★★★★★★★

Basis Adressen für Handler Vektoren für die immer vorhandenen Handler.

SCREEN EDITOR (E) => E400
DISPLAY HANDLER (S) => E410
KEYBOARD HANDLER (K) => E420
PRINTER HANDLER (P) => E430
CASSETTEN HANDLER (C) => E440

★★★★★★★★★★★★★★★★★

OS EINSPRUNG ADRESSEN

★★★ DISKIV 58448 E450 ★★★
Disk Handler Initialisierung

★★★ DISKINV 58451 E453 ★★★
Disk Handler Einsprung

★★★ CIOV 58454 E456 ★★★
CIO Utility Einsprung (oft gebraucht)

★★★ SIOV 58457 E459 ★★★
SIO Utility Einsprung

★★★ SETVBV 58460 E45C ★★★
Routine, um System Timer zu setzen

★★★ SYSVBV 58463 E45F ★★★
Stage 1 VBLANK Einsprung

★★★ XITVBV 58466 E462 ★★★
Einsprung zum Verlassen von VBLANK

★★★ SIOINV 58469 E465 ★★★
SIO Utility Einsprung

★★★ SENDEV 58472 E468 ★★★
Send enable Routine

★★★ INTINV 58475 E46B ★★★
Interrupt Handler Initialisierung

★★★ CIOINV 58478 E4E6 ★★★
CIO Utility Initialisierung

★★★ BLKBDV 58481 E471 ★★★
Blackboard Mode entry

★★★ WARMSV 58484 E474 ★★★
Warmstart Einsprung (RESET)

★★★ COLDSV 58487 E477 ★★★
Kaltstart Einsprung (Einschalten)

★★★★★★★★★★★★★★★★★

Buchtips

ZX Microdrive-Buch

Programme, Maschinencode, Netzwerke. Autor: Andrew Pennell. Verlag: Birkhäuser. 130 Seiten, 27.80 Mark.

Sinclair ZX Spectrum

Programme zum Lernen und Spielen. Aus dem Englischen. 1. Auflage 1983. Autor: Tim Hartnell, Verlag Sybex,224 Seiten, 28 Mark.

Spectrum ohne Grenzen

Führt Schritt für Schritt in die Feinheiten dieses Computers ein. Autor: T. Hartnell/D. Jones. Verlag: Hueber. 208 Seiten, 29.80 Mark.

Der Sinclair Spectrum ROM

Spielebuch für alle ZX-Spectrum-Freunde. Autoren: R. Arenz/M. Görlitz. Verlag: Hueber, 214 Seiten, 39.80 Mark.

Dietrich Freise
Kölner Damm 2
1000 Berlin 47
Tel:030/6049226

G.Schättiger
Am Horner Moor 24
2000 Hamburg 74
Tel:040/65511224

Peter Schneider
Gropiusring 45
2000 Hamburg 60
Tel:040/6314666

Dirk Bannick
Marienhofweg 29
2250 Husum
Tel:04841/72704

Ulrich Schmitz
Heidestr.5
3012 Langenhagen 4
Tel:0511/782693

Holger Kipp
Steige 6
3050 Holzminden
Tel:05531/1647

Fritz Rohde
Berliner Str.3
3160 Lehrte 9
Tel:05175/2691

Detlef Streeck
Pferdemarkt 2
2160 Stade
Tel:04141/2221

Wolfgang Roberts
Heideweg 125
4000 Düsseldorf 30
Tel:0211/613653

Hans Peter Küsters
Bankstr.12
4000 Düsseldorf 30
Tel:0211/4981220

Arnd Petmecky
Wildungerstr.7
4100 Duisburg 25
Tel:0203/789815

Angelo Visintainer
Bittorfstr. 49
4400 Münster
Tel:0251/81889

Michael Schürmann
Rochusstr.343
5000 Köln 30
Tel:0221/591108

C.J. Kammler
Kastanienallee 50
5603 Wulfrath
Tel:02058/70516

Knuth Hermann
Meerscheiderstr.130
5650 Solingen 11
Tel:

C. Quigtar
Am Eisernschlag 27
6000 Frankfurt 50
Tel:0611/518687

Günther Heinz Best
Berliner Str.228
6050 Offenbach
Tel:0611/884541

Reginald Eder
Lessingstr.18
6500 Mainz
Tel:06131/678444

Peter Resch
Weser Str.19
6086 Riedstadt 1
Tel:

Günther Möhle
Ostring 3
6452 Hainburg 1
Tel:06182/69709

Peter E. Stephan
Auguste-Supper-Str.8
7140 Ludwigsburg
Tel:

Jürgen Graßl
Rintheimer Hauptstr.33
7500 Karlsruhe
Tel:0721/613834

Jürgen Rüd
Eisenbahnstr.66
7800 Freiburg
Tel:0761/22012

Harald Zoschke
Kleinhardpenning 7a
8150 Holzkirchen 2
Tel:08024/35920

Roland Wassenberg
Neusser Str. 30
4005 Meerbusch 1
Tel:02105/76156

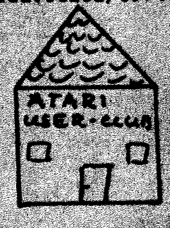
Jürgen Bott
Rohrweg 37
7031 Gärtingen
Tel:

Detlef Gehl
Kirchfeldstr.59
4030 Ratingen 8
Tel:02102/50217

Björn Andreas
Allemannenweg 13
7036 Schönaich
Tel:07031/51637

Ralf Büttner
Gustav-Heinemann-Str.25
5090 Leverkusen
Tel:0214/75333

Klaus Ullmann
Hendelstr.2
6200 Wiesbaden
Tel:02101/522146





In dieser Ausgabe

Kirschen-Karl
Quadratische Gleichungen
Graph
Nullstelle
Morsen lernen
Sternbilder und Planeten
Princess
Nuclear
Danger Snake
Jackpot
Kanopus
Siebzehn und vier
Castle Combat
Töne merken
Der siebte Sinn für Zahlen
Saurer Regen
X und O
Finanzmanager
Sortieren numerischer Felder
Programm-Bibliothek
Versandaufkleber
Datenbank
Laterna Magica
Listformat
Fehler-Melder
Change

Tips zum Erstellen eigener Spiele

Einstieg in Player-Missile-Grafik, Teil 1 und 2
Die wichtigsten Poke-Befehle
Hilfe für Selbstprogrammierer